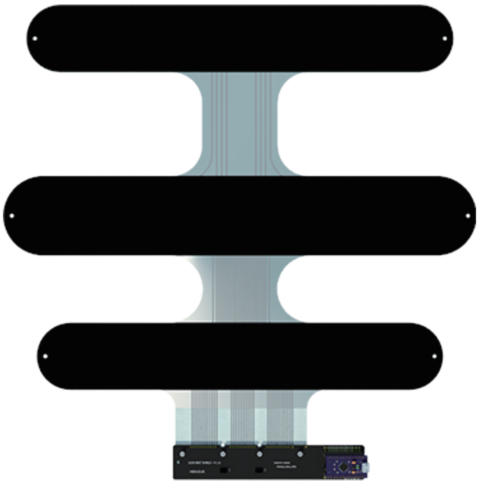
# Manual - Cushion sensor coding kit

## Model : Cushion sensor coding kit



Cushion sensor coding kit

## Introduction

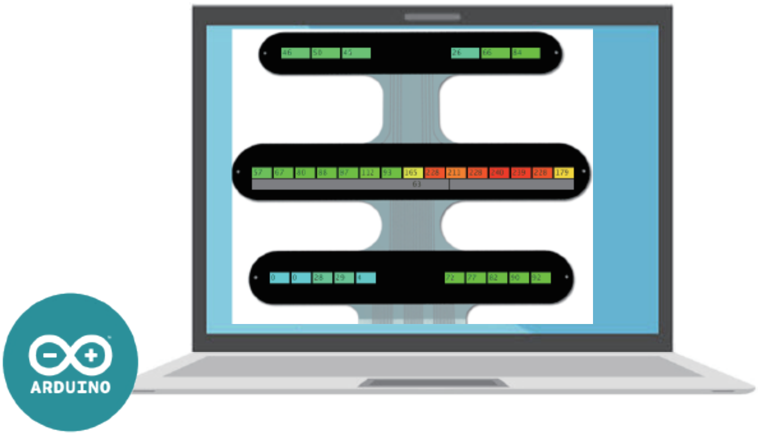
This product is a measurement kit that collects the pressure distribution data of a seated person on a chair.

The product composition is as follows. (See page 4)
(1) Cushion type pressure sensor
(2) Arduino for measurement
(3) Arduino shield for connecting and measuring sensors and circuits

The cushion sensor consists of 31 individual pressure sensors.
You can get information about the sitting posture from the measured values at 31 positions.

This product provides the basic H/W and S/W necessary to measure and show data.
You can develop various application solutions using the measured data.

< Challengeable applications >
*This kit can be used*
-To develop healthcare solution : Monitor sitting posture to warn bad posture
-To develop input device : With the tilting information, you can generate a direction key events and utilizes it as a VR (Virtual Reality) or game controller.
-To design ergonomic cushion/chair
-To develop for the disabled



Seated posture measured with this kit and displayed on a PC screen
SW development environment: Arduino, Processing

## How to use

In addition to the components of this product, the user must additionally prepare a USB cable, a PC, and a simple cushion (or thick cloth if there is no cushion). This sensor is used inside/under a seat cushion. There are various methods to measure the output of the cushion sensor, but in this document, it is measured through the Arduino circuit. It provides basic source code for Arduino and PC, and users can modify the source code.

Precaution is at the end of this document.

## Reference documents / sites 🔗

It is possible to understand basic principles and implement with only this document. Additional resources or examples can be found at the link below.
Go to the document library: https://bit.ly/2WliDm4
Source code example blog: https://bit.ly/2ZwEkBM

# Cushion sensor recognition examples

## Model : Cushion sensor coding kit

### Data usage example

You can understand the tendency of the sitting posture through the measurement values of the 31 cells in the cushion sensor. The measuring range of one sensor cell is about 20gf to 4Kgf. The actual load applied to one sensor is usually within 3 Kgf, as the load of the seated person spreads across the whole chair and the load is distributed due to the softness of the cushion.

For this reason, even if the load of the seated person is as high as 130kg, pressure measurement and posture recognition are possible.

Recognizable seating poses include left/right tilt, degree of tilt, legs-crossed etc.

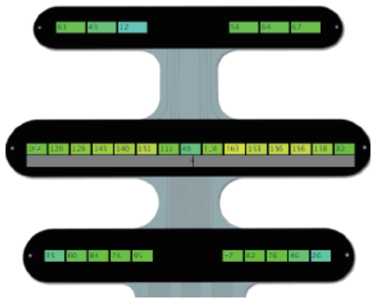
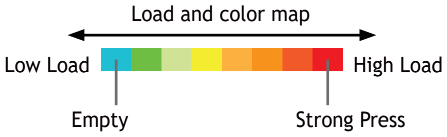
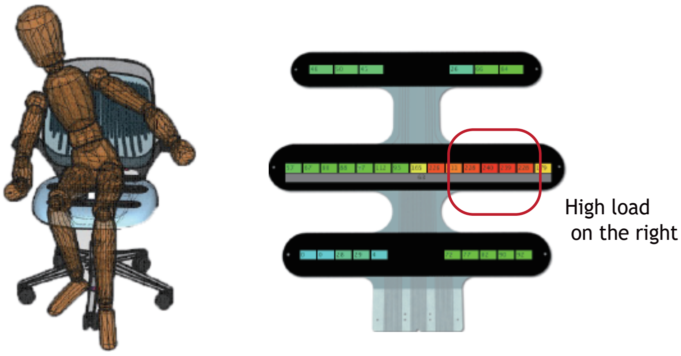


Figure) Measured values in normal posture.
Most of the 31 cells look light green and yellow.

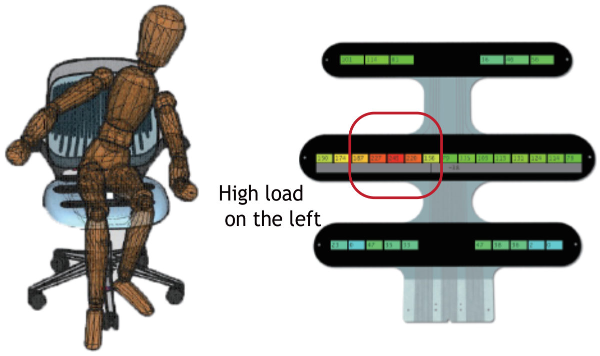
Coding for Measurement :
see pages 6-17 of this document



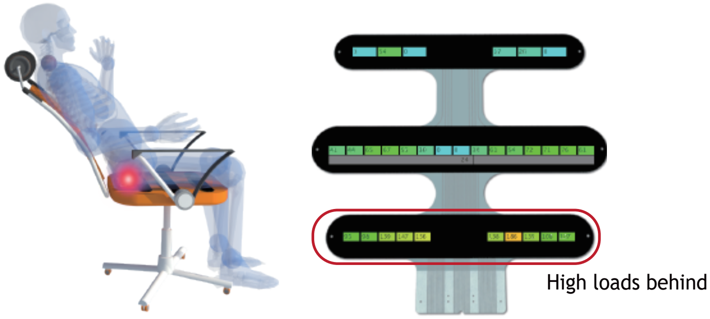
### Right tilted posture and measurement results



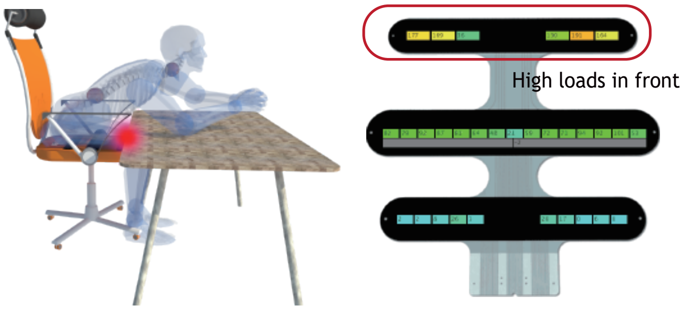
### Left tilted posture and measurement results



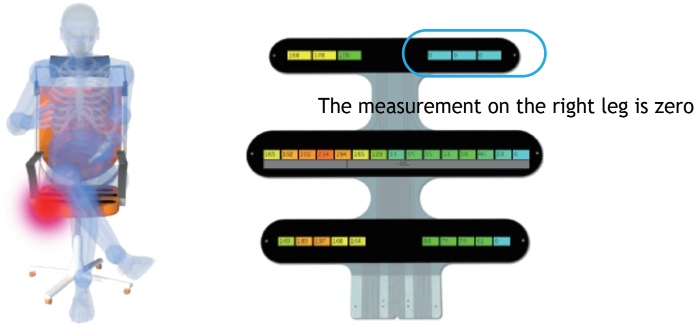
### Back tilted posture and measurement results



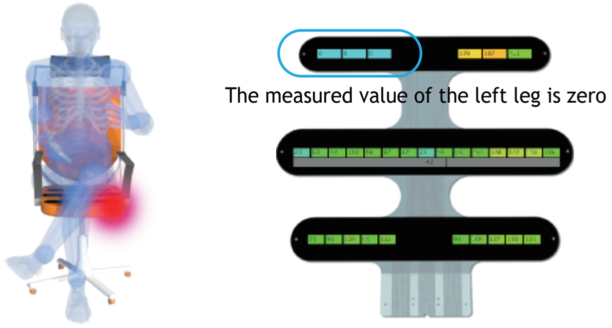
### Forward tilted posture and measurement results



### Posture with crossing right leg



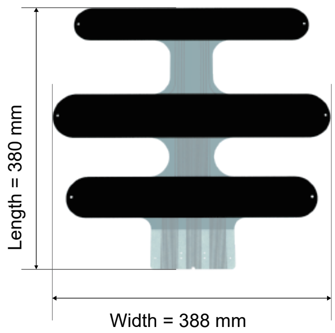
### Posture with crossing left leg

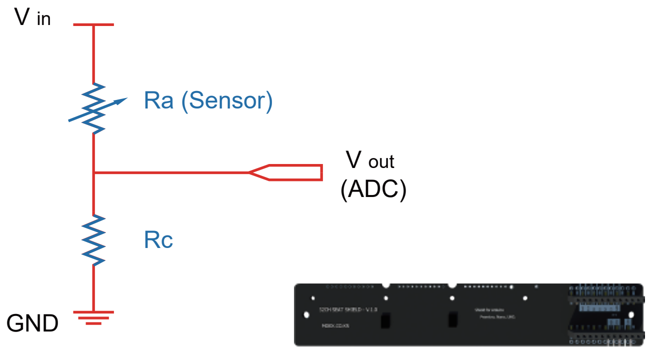


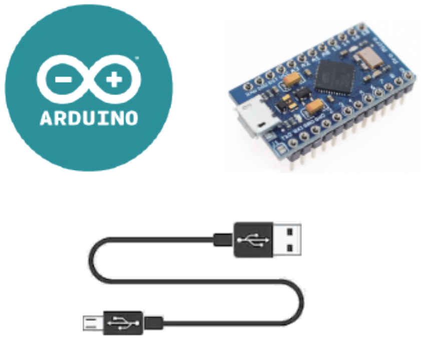
There may be an error in the attitude recognized/determined by this cushion sensor. Factors affecting: siting body shape, chair/cushion shape, material, etc.

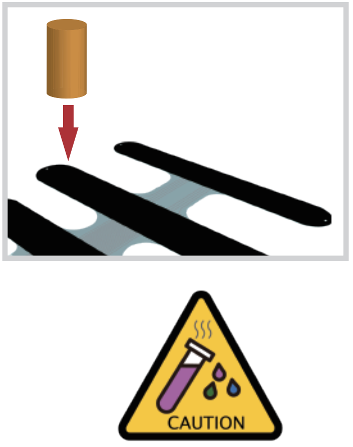
# Table of Contents

## Model : Cushion sensor coding kit





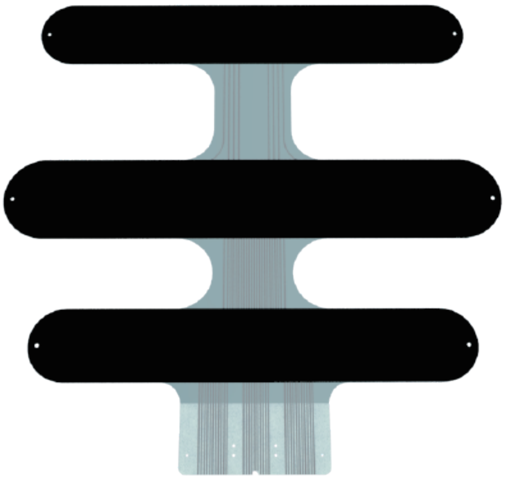




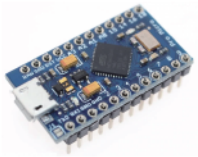
# Connecting the cushion sensor to the Arduino

## Model : Cushion sensor coding kit

## Kit component



Cushion sensor (Model : MDXS-16-5610)



Name: Arduino Pro Micro (or Nano)
Role: Measure the cushion sensor 31 channels.

Be sure to install the driver before connecting the USB to the PC. (See pages 10 and 11)

Name: Cushion Sensor Shield
Role: Role : To connect sensor and select a cell

Two analog switches and pulldown resistor for ADC measurement are included inside.

Accessories for assembly

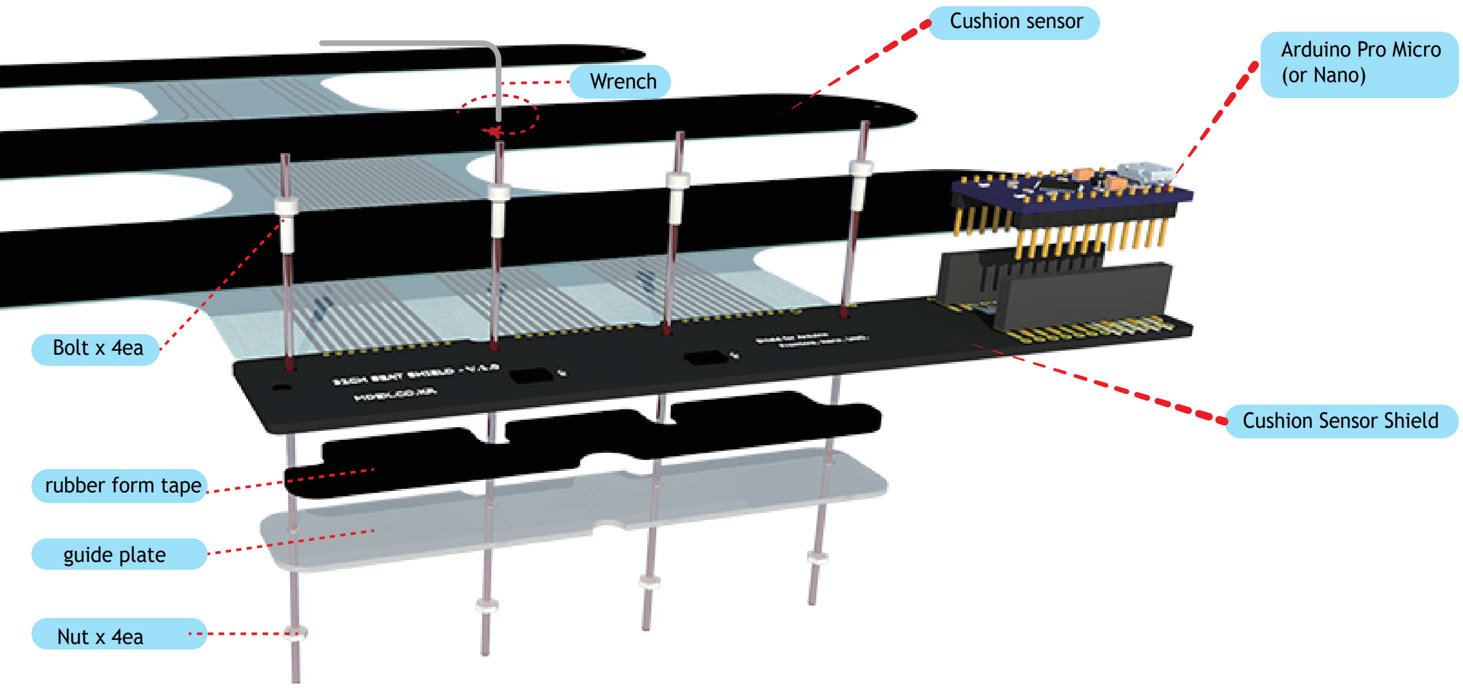
Bolt x 4ea
Nut x 4ea

Wrench

rubber form tape

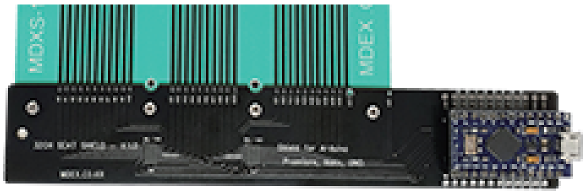
guide plate

## Assembly-cushion sensor, shield, arduino

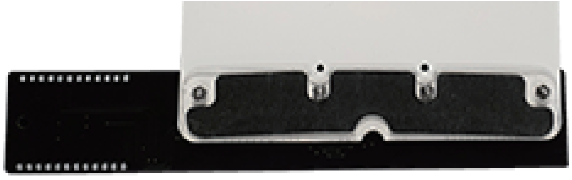


## After tightening the bolt.

Top view



Bottom view

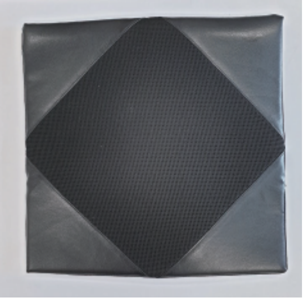


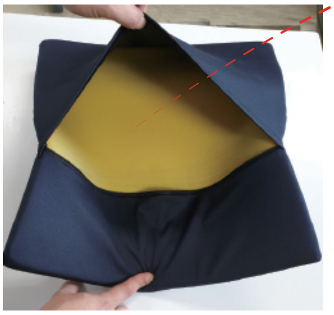
# Put the sensor into the cushion

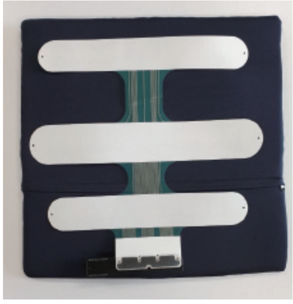
Model : Cushion sensor coding kit

## Insertion sequence

1

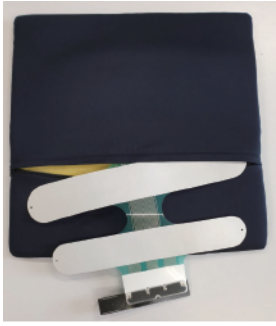

Cushion top

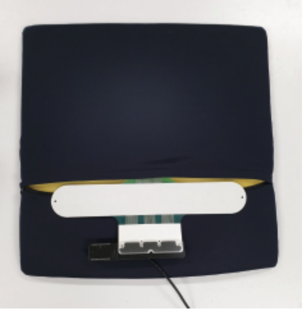

Turn the cushion upside down, and open the zipper.

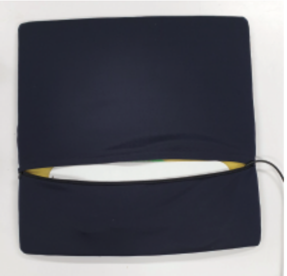

Size comparison-sensor and cushion

The material inside the cushion should be soft, such as sponge, memory foam, and cotton.

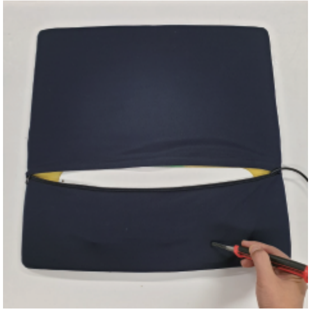
2

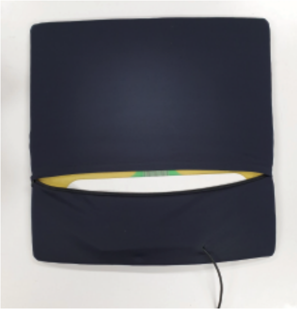

Open the zipper and insert the sensor upside-down as you see.

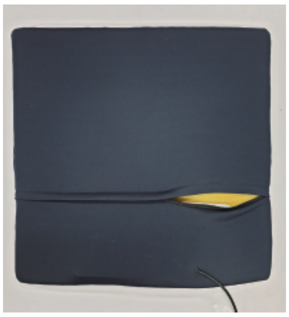

Connect the Micro USB cable to the shield.


Close the zipper after putting it all in.

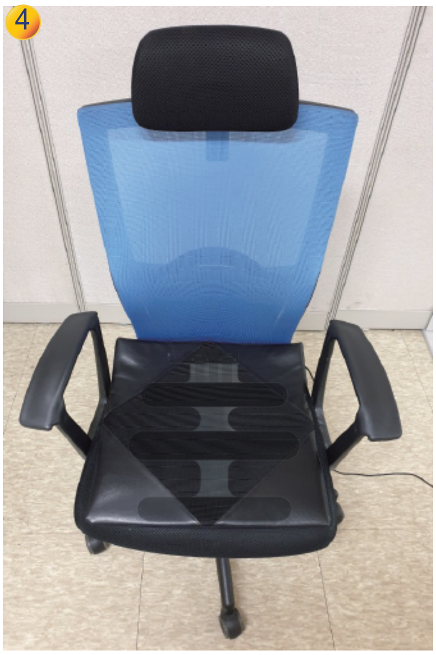
3


If the cable folds inside the cushion, it may cause a measurement error. Make a small hole on the cover.

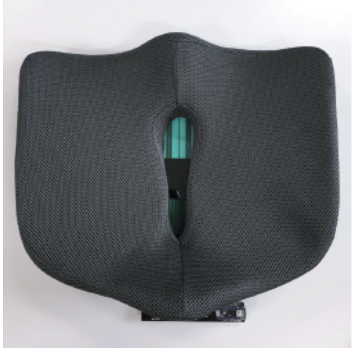

Pull out USB cable through a hole

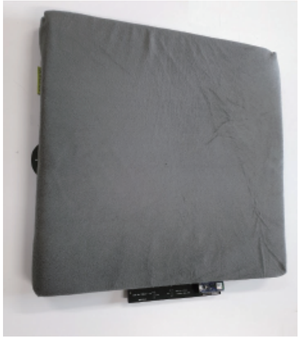

Close the zipper. Done!

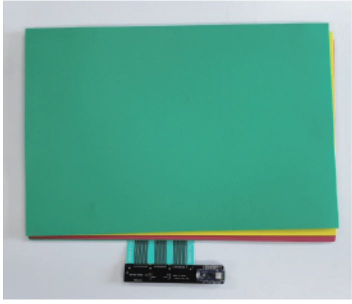
4


Turn the cushion over again and place it on the chair with the top side up.
Sit down now.

It is not necessary to put the sensor in the cushion. You can also place the sensor under the cushion. However, make sure that the hard and protruding parts, such as zippers, do not press the sensor's measurement area.


Healthcare cushion


Cushion smaller than the sensor


fabric/foam/blanket

# Sensor dimension, Cell placement

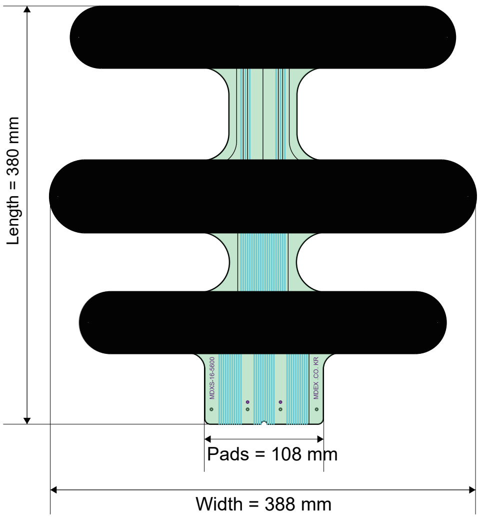
## Model : Cushion sensor coding kit

## Sensor dimensions

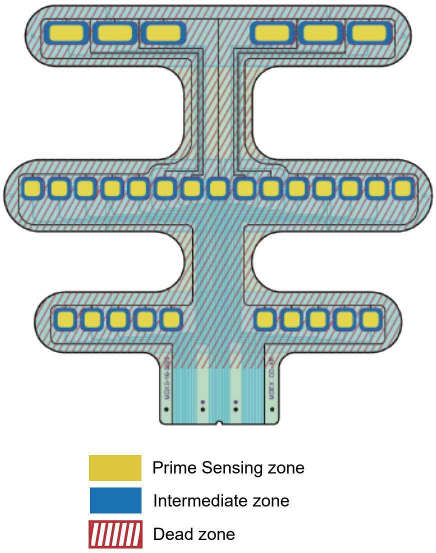
### Summary

| Unit | Description |
|---|---|
| Length | 380mm |
| Width | 388mm |
| Thickness | 0.95mm |
| Width of 1 cell | 19 or 38mm |
| Length of 1 cell | 19mm |
| Sensing range (per cell) | 20gf ~ 4Kgf |
| Number of cells | 31 |

Top view



Sensing zone



## About Cell Placement

Let's find out why we placed 31 sensors in three rows and why we placed more cells in the middle row of three rows. The occupant's load is widely distributed, but not entire the chair.
The load is intensively applied mainly around the hip bone. (Picture on the right) For this reason, the pressure distribution of the occupant is usually concentrated in the position of the hip bone, as shown in the left figure below.
The load applied to the thigh area is relatively low.

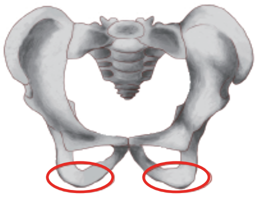


Figure) The load is intensively applied to the two hip bones.

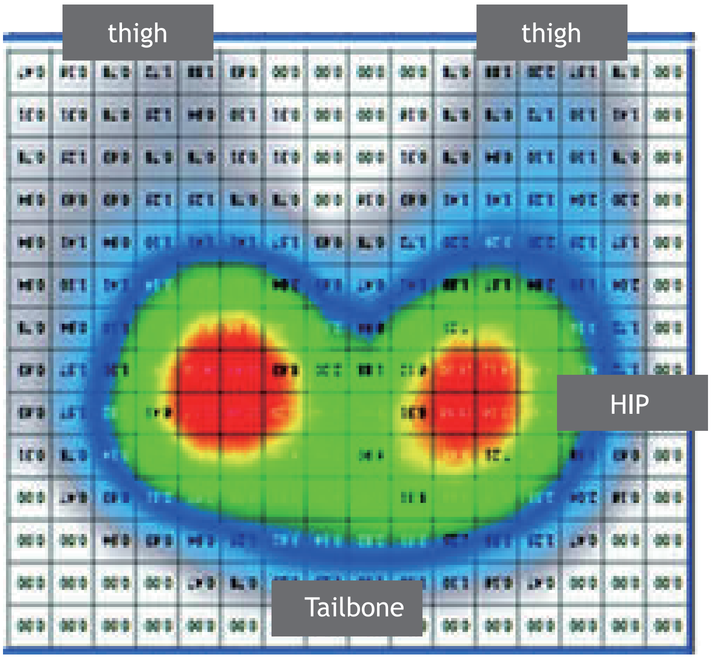


Figure) Typical load distribution of seated persons.

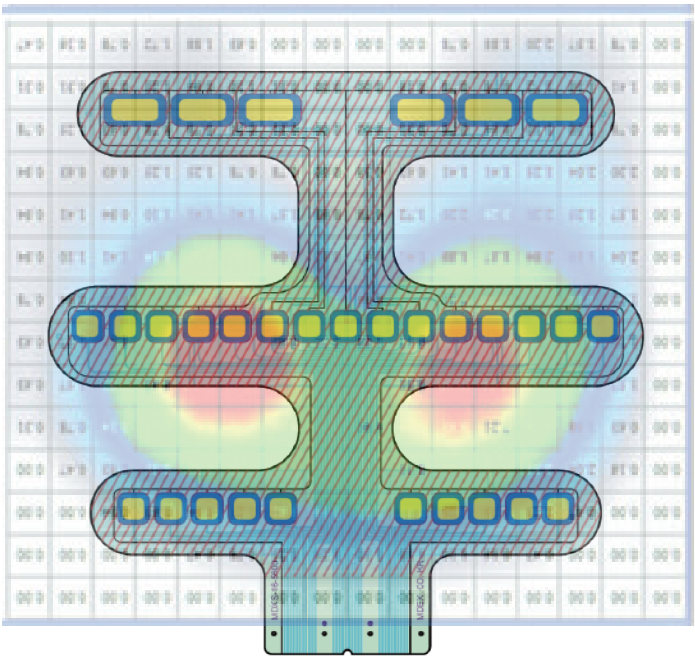


Figure) siting load distribution and cell placement. A lot of sensors placed in the hip area. The sensor on the thigh detects whether the thigh has touched the chair

# Circuit) Shield for Arduino and terminal

Model : Cushion sensor coding kit

## Composition of cushion sensor shield-socket, pad for sensor connection, analog switch

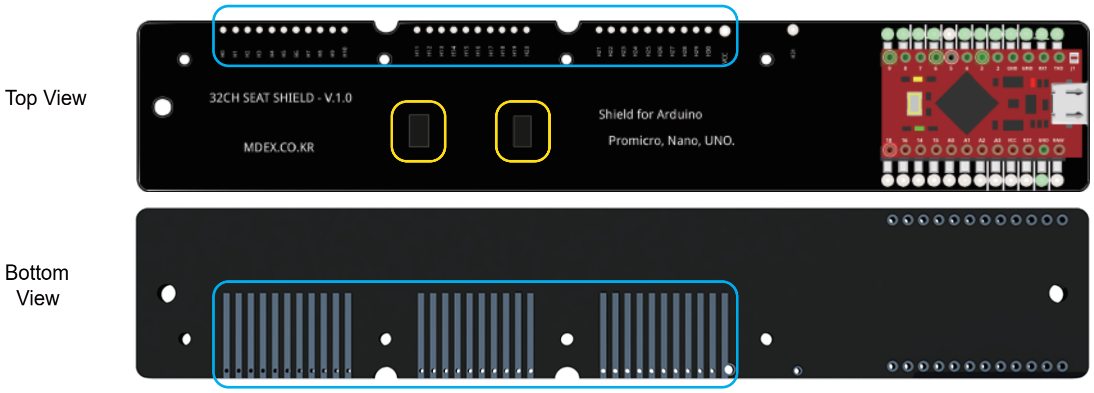
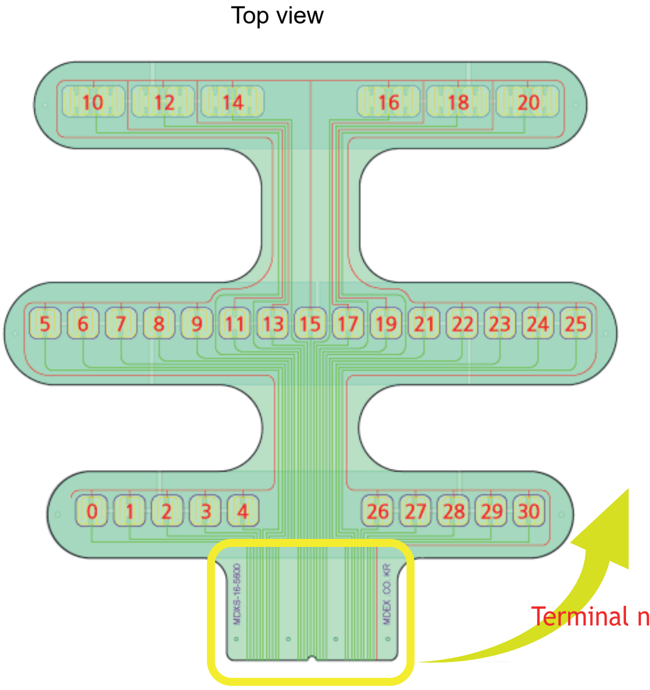


Figure) Arduino Promicro plugged into the socket of the cushion sensor shield. (Arduino genuine is red while compatible is blue)

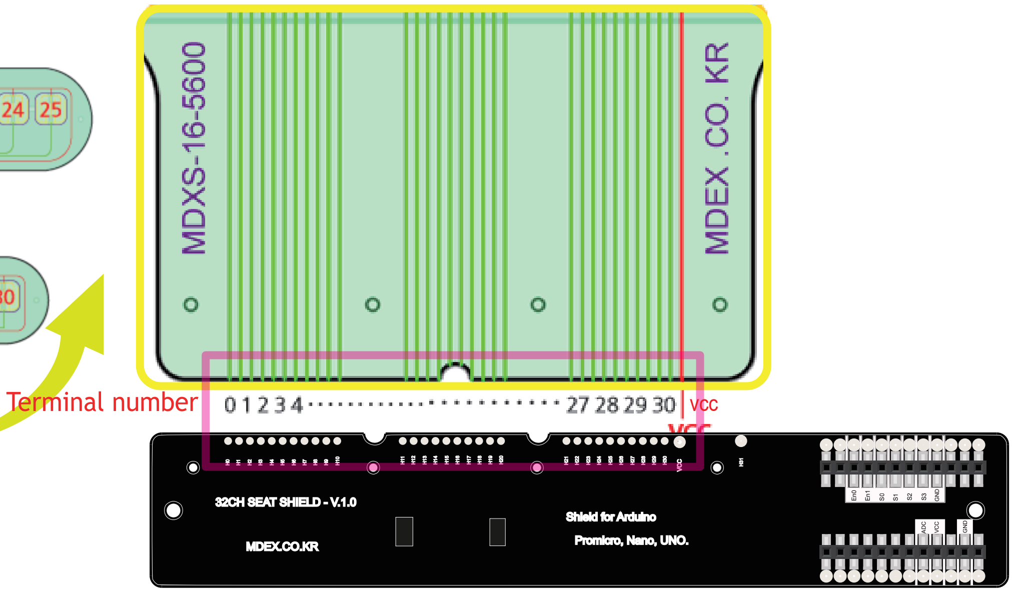
🟦 : Cushion sensor connection / 32 fixing pads

🟨 : 2 ea of built-in 4to16 analog switches

## Number of 31 cells (0~30) and data transmission protocol



The sensor numbers on the left are connected according to the terminal numbers (0 to 30) in the figure below.
The sensor measurement values are transmitted in the order of the terminal number.



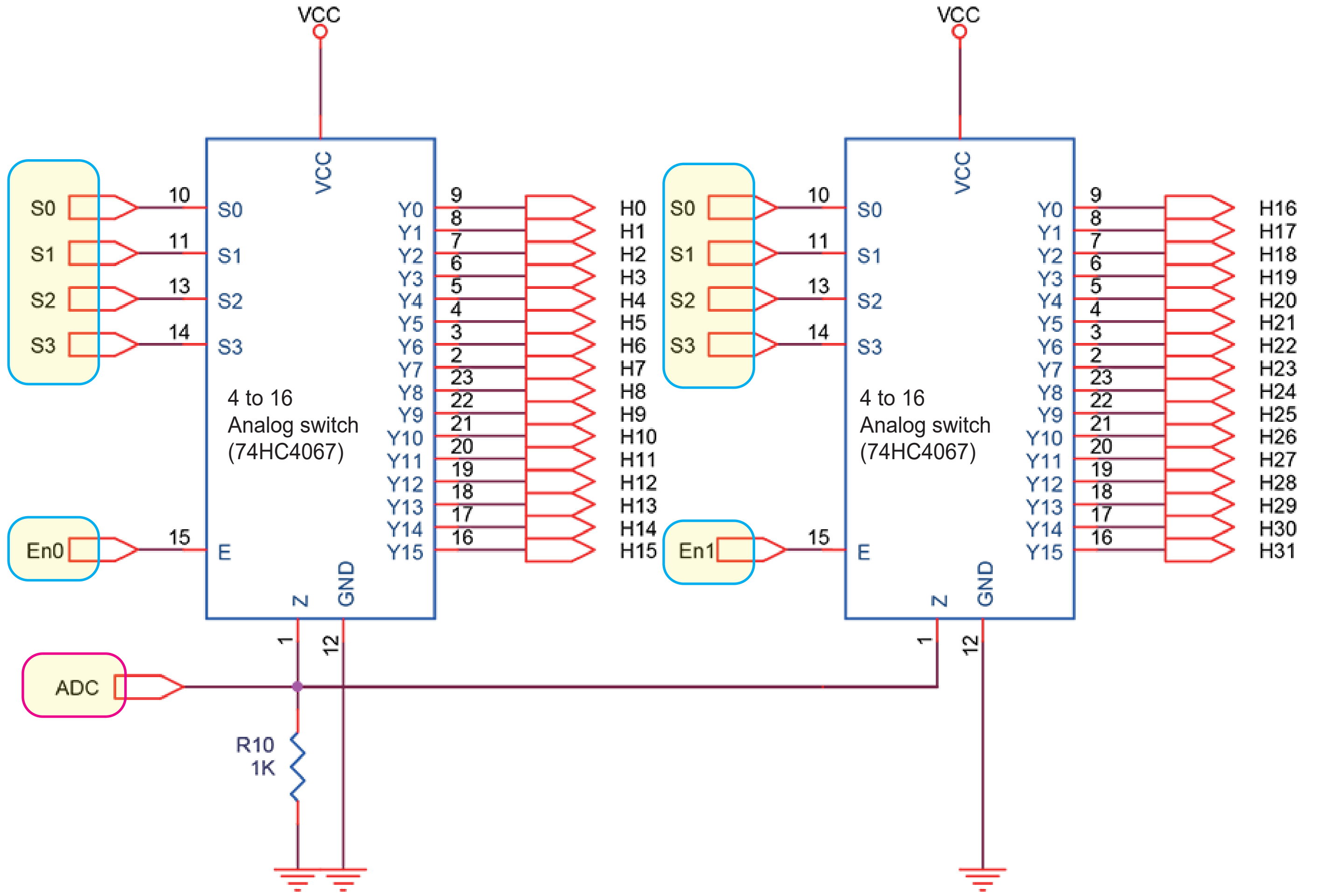
Cushion sensor shield for Arduino

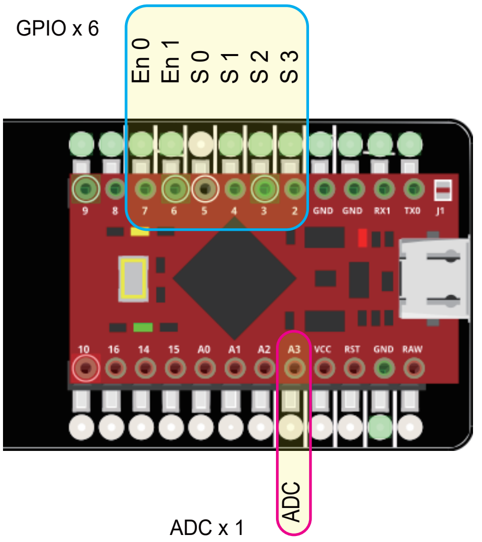
# Circuit) Sensor shield - 1

Model : Cushion sensor coding kit

## Schematic of cushion sensor shield

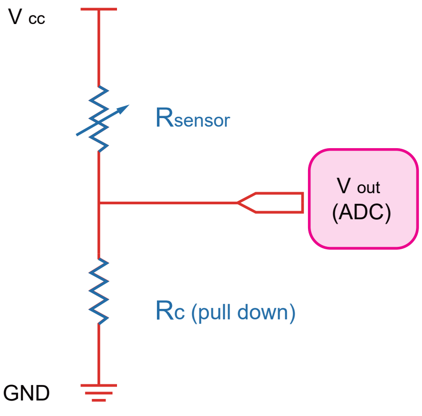
Two analog switches (74HC4067; 4 to 16) are built in the cushion sensor shield.
32 sensors (H0~H31 below) can be connected by 6 GPIOs (S0 ~ S3, En0, En1) below).





## Measuring 1 sensor cell -ADC circuit

One ADC terminal and one pull-down resistor are required.

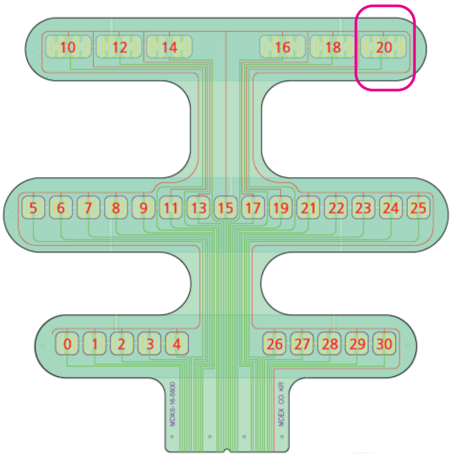


$$V_{out} = \frac{R_c}{R_{sensor} + R_c} \times V_{cc}$$

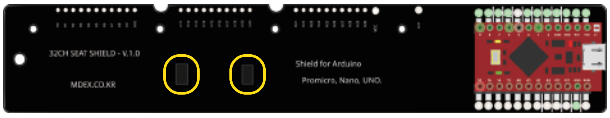
# Circuit) Sensor shield - 2

## Model : Cushion sensor coding kit

### Cell selection example - 20th cell



74HC4067 Switch Description. Connect 16 cells with 4 signal pins.



Selecting the 20th cell in the cushion sensor shield

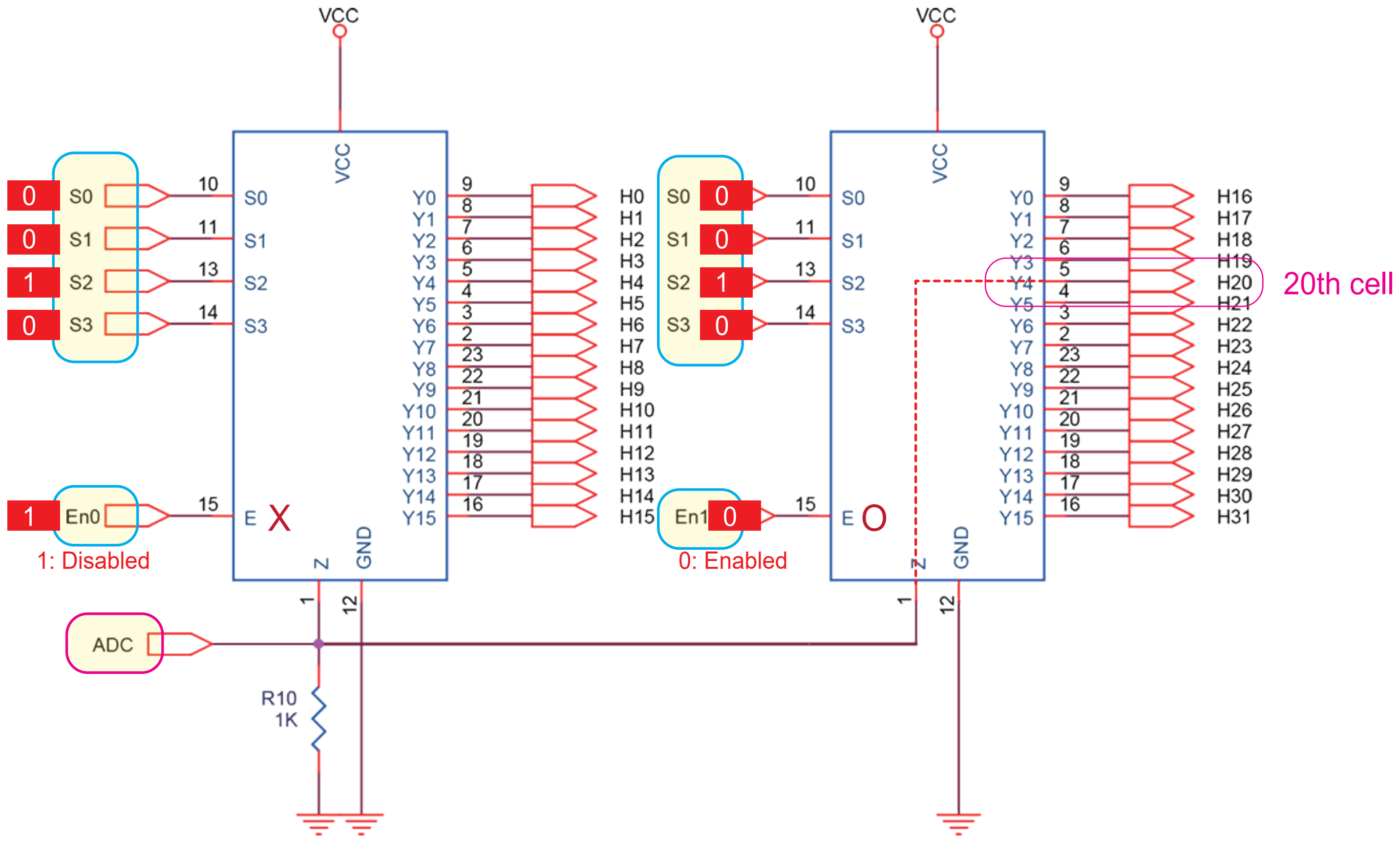
| | En 0 | En 1 | S0 | S1 | S2 | S3 |
|---|---|---|---|---|---|---|
| H0 | 0 | 1 | 0 | 0 | 0 | 0 |
| H1 | 0 | 1 | 1 | 0 | 0 | 0 |
| H2 | 0 | 1 | 0 | 1 | 0 | 0 |
| H3 | 0 | 1 | 1 | 1 | 0 | 0 |
| H4 | 0 | 1 | 0 | 0 | 1 | 0 |
| H5 | 0 | 1 | 1 | 0 | 1 | 0 |
| H6 | 0 | 1 | 0 | 1 | 1 | 0 |
| H7 | 0 | 1 | 1 | 1 | 1 | 0 |
| H8 | 0 | 1 | 0 | 0 | 0 | 1 |
| H9 | 0 | 1 | 1 | 0 | 0 | 1 |
| H10 | 0 | 1 | 0 | 1 | 0 | 1 |
| H11 | 0 | 1 | 1 | 1 | 0 | 1 |
| H12 | 0 | 1 | 0 | 0 | 1 | 1 |
| H13 | 0 | 1 | 1 | 0 | 1 | 1 |
| H14 | 0 | 1 | 0 | 1 | 1 | 1 |
| H15 | 0 | 1 | 1 | 1 | 1 | 1 |
| H16 | 1 | 0 | 0 | 0 | 0 | 0 |
| H17 | 1 | 0 | 1 | 0 | 0 | 0 |
| H18 | 1 | 0 | 0 | 1 | 0 | 0 |
| H19 | 1 | 0 | 1 | 1 | 0 | 0 |
| H20 | 1 | 0 | 0 | 0 | 1 | 0 |
| H21 | 1 | 0 | 1 | 0 | 1 | 0 |
| H22 | 1 | 0 | 0 | 1 | 1 | 0 |
| H23 | 1 | 0 | 1 | 1 | 1 | 0 |
| H24 | 1 | 0 | 0 | 0 | 0 | 1 |
| H25 | 1 | 0 | 1 | 0 | 0 | 1 |
| H26 | 1 | 0 | 0 | 1 | 0 | 1 |
| H27 | 1 | 0 | 1 | 1 | 0 | 1 |
| H28 | 1 | 0 | 0 | 0 | 1 | 1 |
| H29 | 1 | 0 | 1 | 0 | 1 | 1 |
| H30 | 1 | 0 | 0 | 1 | 1 | 1 |
| H31 | 1 | 0 | 1 | 1 | 1 | 1 |

Table) How to set GPIO pin by cushion cell number
En : Enable Negative
S0~S3 : Signal

( En0=1, En1=0, S0=0, S1=0, S2=1, S3=0 )

Pin setting to select 20th cell

### ✓ Ex) How to set 6 pins to select the 20th cell



# Connect Arduino to PC

Model : Cushion sensor coding kit

## Arduino setup & configure

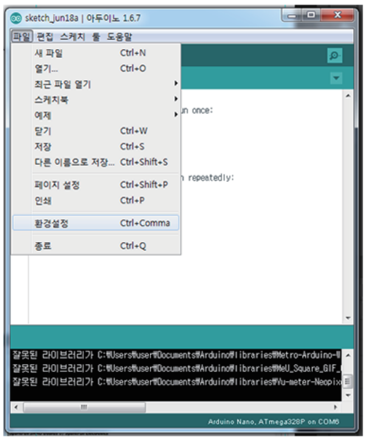
Link to Arduino installation file - https://www.arduino.cc/en/Main/Software

If you are using Arduino Nano or Uno, the following steps are not necessary.

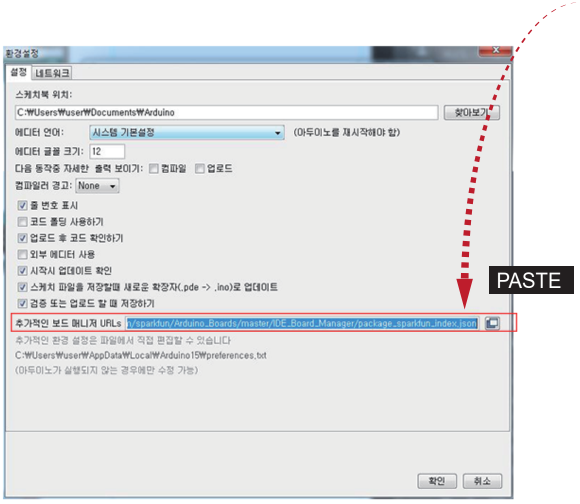
## Arduino Configuration - Adding an Arduino Pro Micro Board

File> Preferences> Settings tab and copy/paste the following address into the ‘Additional Boards Manager URL’s field.

COPY https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json



```
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Metro-Arduino-
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩MeU_Square_GIF_
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Vu-meter-Neopix
```



스케치북 위치: C:₩Users₩user₩Documents₩Arduino

에디터 언어: 시스템 기본설정 (아두이노를 재시작해야 함)

에디터 글꼴 크기: 12

다음 동작중 자세한 출력 보이기: □ 컴파일 □ 업로드

컴파일러 경고: None

☑ 줄 번호 표시

□ 코드 폴딩 사용하기

☑ 업로드 후 코드 확인하기

□ 외부 에디터 사용

☑ 시작시 업데이트 확인

☑ 스케치 파일을 저장할때 새로운 확장자(.pde -> .ino)로 업데이트

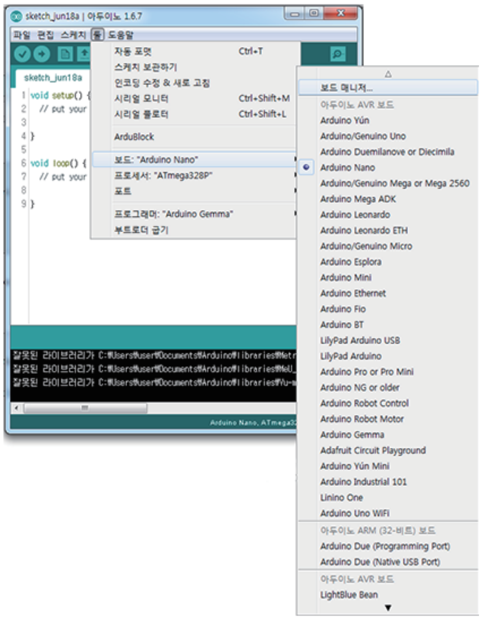
☑ 검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs: n/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:₩Users₩user₩AppData₩Local₩Arduino15₩preferences.txt
(아두이노가 실행되지 않는 경우에만 수정 가능)

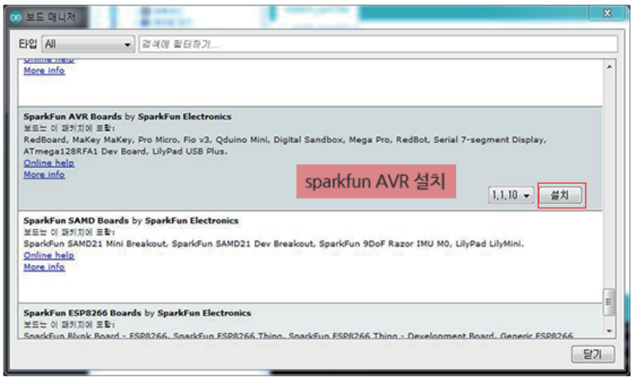
## Install Arduino Pro Micro Package

Tools> Board> Board manager> Search for'sparkfun AVR' in the search bar> Click the'Install' button



```
sketch_jun18a
void setup() {
  // put your
}

void loop() {
  // put your
}
```

```
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Metr
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩MeU
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Vu-
```



More info

**SparkFun AVR Boards** by **SparkFun Electronics**
보드는 이 패키지에 포함:
RedBoard, MaKey MaKey, Pro Micro, Fio v3, Qduino Mini, Digital Sandbox, Mega Pro, RedBot, Serial 7-segment Display, ATmega128RFA1 Dev Board, LilyPad USB Plus.
Online help
More info

**SparkFun SAMD Boards** by **SparkFun Electronics**
보드는 이 패키지에 포함:
SparkFun SAMD21 Mini Breakout, SparkFun SAMD21 Dev Breakout, SparkFun 9DoF Razor IMU M0, LilyPad LilyMini.
Online help
More info

**SparkFun ESP8266 Boards** by **SparkFun Electronics**
보드는 이 패키지에 포함:
SparkFun Blynk Board - ESP8266, SparkFun ESP8266 Thing, SparkFun ESP8266 Thing - Development Board, Generic ESP8266

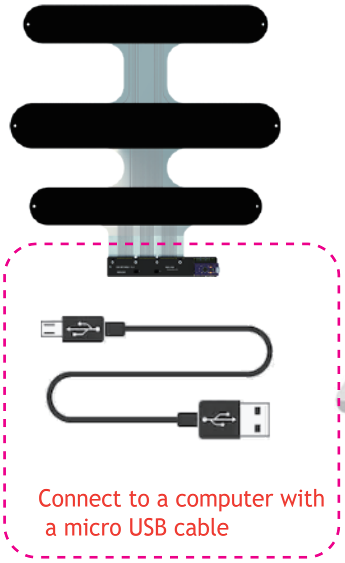
Install sparkfun AVR in Board manager

# Connect Arduino to PC

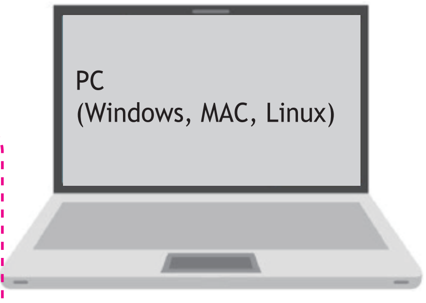
Model : Cushion sensor coding kit

## Connect Arduino Pro Micro to PC-USB

Cushion sensor coding kit



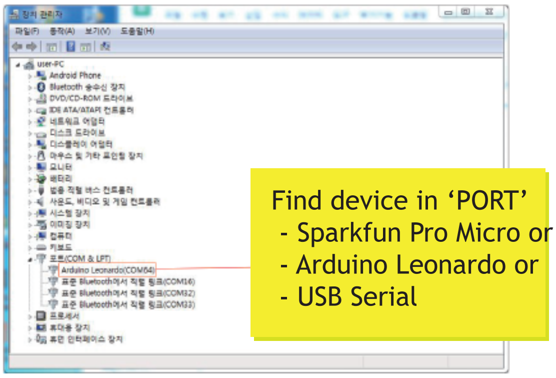
Connect to a computer with a micro USB cable



For Arduino compatible boards using the CH340 chip, the CH340 driver must be installed.

## Device Recognition Check-Device Manager

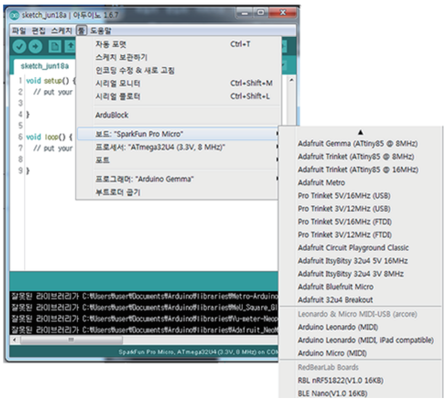
Settings> Device Manager> Port (COM & LPT)
Check if a device with one of the names below is found



Find device in ‘PORT’
- Sparkfun Pro Micro or
- Arduino Leonardo or
- USB Serial

## Arduino Board selection

1) Selecting a board
--> Sparkfun Pro Micro Or Arduino Nano

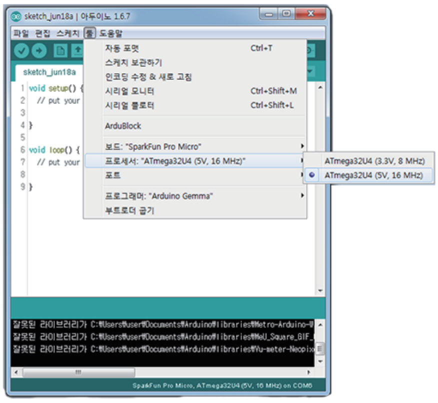

sketch_jun18a | 아두이노 1.6.7
파일 편집 스케치 툴 도움말
자동 포맷 Ctrl+T
스케치 보관하기
인코딩 수정 & 새로 고침
시리얼 모니터 Ctrl+Shift+M
시리얼 플로터 Ctrl+Shift+L
ArduBlock
보드: "SparkFun Pro Micro"
프로세서: "ATmega32U4 (3.3V, 8 MHz)"
포트
프로그래머: "Arduino Gemma"
부트로더 굽기
Adafruit Gemma (ATtiny85 @ 8MHz)
Adafruit Trinket (ATtiny85 @ 8MHz)
Adafruit Trinket (ATtiny85 @ 16MHz)
Adafruit Metro
Pro Trinket 5V/16MHz (USB)
Pro Trinket 3V/12MHz (USB)
Pro Trinket 5V/16MHz (FTDI)
Pro Trinket 3V/12MHz (FTDI)
Adafruit Circuit Playground Classic
Adafruit ItsyBitsy 32u4 5V 16MHz
Adafruit ItsyBitsy 32u4 3V 8MHz
Adafruit Bluefruit Micro
Adafruit 32u4 Breakout
Leonardo & Micro MIDI-USB (arcore)
Arduino Leonardo (MIDI)
Arduino Leonardo (MIDI, iPad compatible)
Arduino Micro (MIDI)
RedBearLab Boards
RBL nRF51822(V1.0 16KB)
BLE Nano(V1.0 16KB)

```
void setup() {
  // put your

}

void loop() {
  // put your

}
```

```
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Metro-Arduino
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩MeU_Square_GI
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Vu-meter-Neop
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Adafruit_Neo
```

SparkFun Pro Micro, ATmega32U4 (3.3V, 8 MHz) on COM

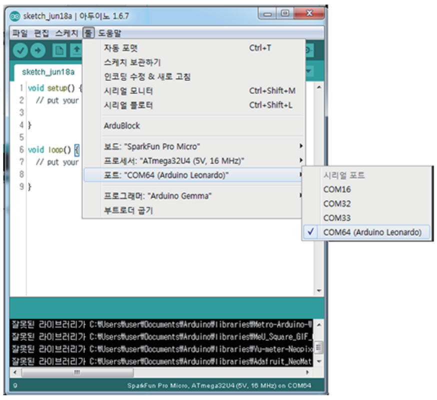
2) Choose Processor (when Arduino Pro Micro)
--> 5V, 16MHz ( O )
--> 3.3V, 8MHz (X)


sketch_jun18a | 아두이노 1.6.7
파일 편집 스케치 툴 도움말
자동 포맷 Ctrl+T
스케치 보관하기
인코딩 수정 & 새로 고침
시리얼 모니터 Ctrl+Shift+M
시리얼 플로터 Ctrl+Shift+L
ArduBlock
보드: "SparkFun Pro Micro"
프로세서: "ATmega32U4 (5V, 16 MHz)"
포트
프로그래머: "Arduino Gemma"
부트로더 굽기
ATmega32U4 (3.3V, 8 MHz)
● ATmega32U4 (5V, 16 MHz)

```
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Metro-Arduino-W
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩MeU_Square_GIF_
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Vu-meter-Neopix
```

SparkFun Pro Micro, ATmega32U4 (5V, 16 MHz) on COM6

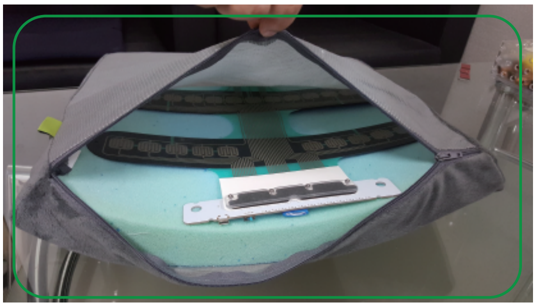
3) Choose Port


sketch_jun18a | 아두이노 1.6.7
파일 편집 스케치 툴 도움말
자동 포맷 Ctrl+T
스케치 보관하기
인코딩 수정 & 새로 고침
시리얼 모니터 Ctrl+Shift+M
시리얼 플로터 Ctrl+Shift+L
ArduBlock
보드: "SparkFun Pro Micro"
프로세서: "ATmega32U4 (5V, 16 MHz)"
포트: "COM64 (Arduino Leonardo)"
프로그래머: "Arduino Gemma"
부트로더 굽기
시리얼 포트
COM16
COM32
COM33
✓ COM64 (Arduino Leonardo)

```
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Metro-Arduino-W
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩MeU_Square_GIF_
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Vu-meter-Neopix
잘못된 라이브러리가 C:₩Users₩user₩Documents₩Arduino₩libraries₩Adafruit_NeoMat
```

SparkFun Pro Micro, ATmega32U4 (5V, 16 MHz) on COM64

Now you are ready to develop.
Please protect the sensor with the cushion before you sit down.

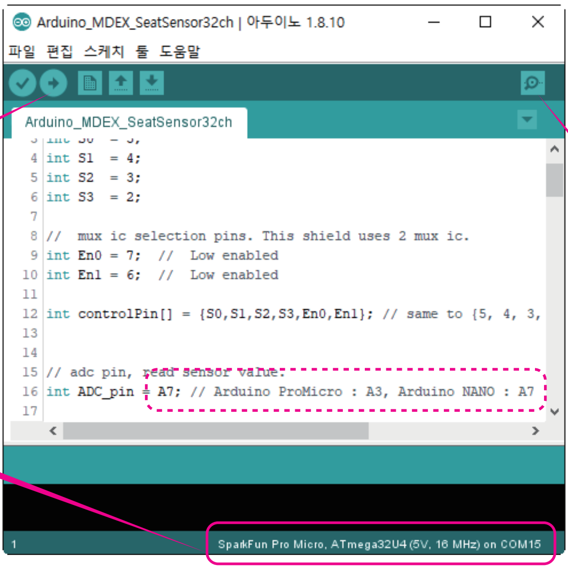


# Measurement example

Model : Cushion sensor coding kit

## Upload/build Arduino source code

Link to the source code example - https://bit.ly/30d0UhS (at the bottom of the shopping mall product page)
After setting the'Arduino Sketch' on the previous page,'Upload' the source code to the Arduino.


Arduino_MDEX_SeatSensor32ch | 아두이노 1.8.10
파일 편집 스케치 툴 도움말
Arduino_MDEX_SeatSensor32ch

```
int S0  = 5;
int S1  = 4;
int S2  = 3;
int S3  = 2;

//  mux ic selection pins. This shield uses 2 mux ic.
int En0 = 7;  //  Low enabled
int En1 = 6;  //  Low enabled

int controlPin[] = {S0,S1,S2,S3,En0,En1}; // same to {5, 4, 3,


// adc pin, read sensor value:
int ADC_pin = A7; // Arduino ProMicro : A3, Arduino NANO : A7

```

SparkFun Pro Micro, ATmega32U4 (5V, 16 MHz) on COM15

2 Source code upload

3 Run Serial Monitor.
See the measured value.

ADC_pin  is A3 if ProMicro and A7 if NANO

Check the sketch settings. 1
Correct board name?
If Pro Micro, 5V, 16Mhz?
Correct COM port number?

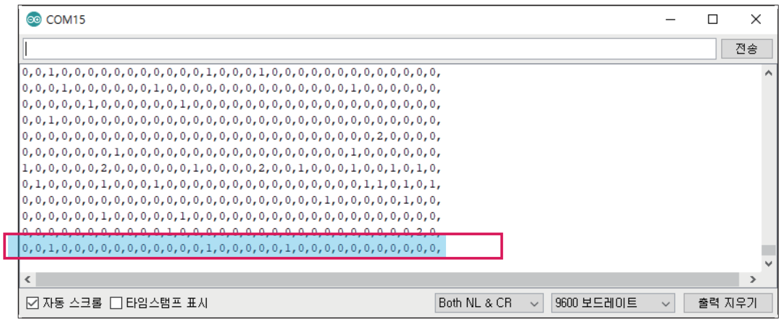
Link to Source Code Description Blog
https://bit.ly/3hXZX4W

## Checking the measured value

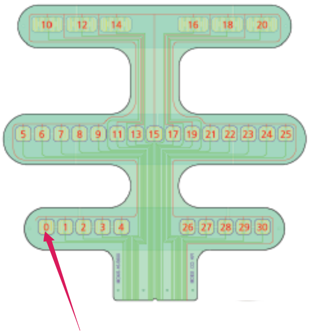
In Menu, Arduino> Tools> Serial Monitor

1) No pressure on the cushion sensor, output values are usually '0' to '5'.

Q) If some low values are shown while no pressure?
A) Check the screw tightness between the sensor and the shield.

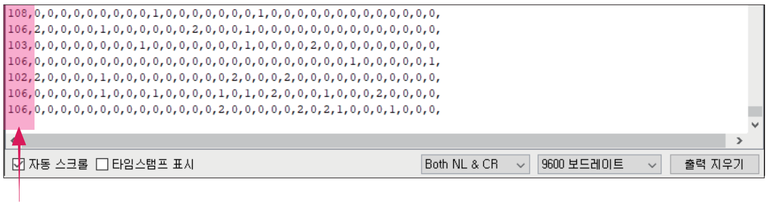


```
0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
1,0,0,0,0,0,2,0,0,0,0,0,0,1,0,0,0,0,2,0,0,1,0,0,0,1,0,0,1,0,1,0,
0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,
0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,
0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
```



0th Cell

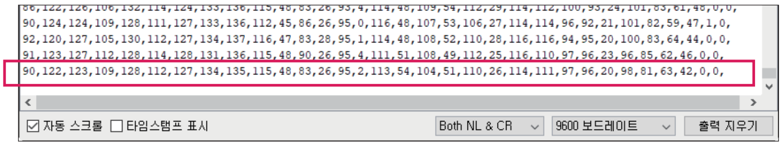
2) Press the 0th cell with your finger. If the 0th value changes, the connection is OK.



```
108,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
106,2,0,0,0,0,1,0,0,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
103,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,0,0,0,0,0,0,0,
106,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,
102,2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,
106,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,2,0,0,0,1,0,0,0,2,0,0,0,0,
106,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,2,1,0,0,0,1,0,0,0,
```

0th Cell

3) A screen while sitting on the cushion sensor.
Every cell value changes.



```
86,122,126,106,132,114,124,133,136,115,48,83,26,93,4,114,48,109,54,112,29,114,112,100,93,24,101,83,61,48,0,0,
90,124,124,109,128,111,127,133,136,112,45,86,26,95,0,116,48,107,53,106,27,114,114,96,92,21,101,82,59,47,1,0,
92,120,127,105,130,112,127,134,137,116,47,83,28,95,1,114,48,108,52,110,28,116,116,94,95,20,100,83,64,44,0,0,
91,123,127,112,128,114,128,131,136,115,48,90,26,95,4,111,51,108,49,112,25,116,110,97,96,23,96,85,62,46,0,0,
90,122,123,109,128,112,127,134,135,115,48,83,26,95,2,113,54,104,51,110,26,114,111,97,96,20,98,81,63,42,0,0,
```

# Visualization SW - 1) 3 rows

Model : Cushion sensor coding kit

## Setup Processing

Link to setup Processing - https://processing.org/download/

Link to source code for Processing - https://bit.ly/2ZwEkBM

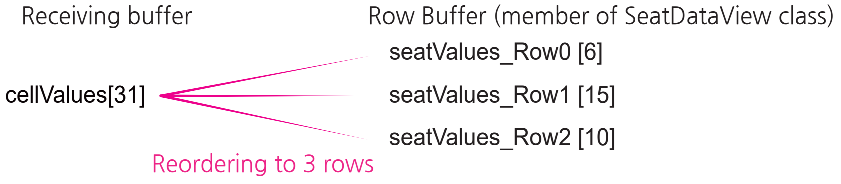
## Data protocol and reordering

1) Reading data, receiving buffer

Function : readSerial_MDEX32chShield()
The Arduino board reads 32 data from the computer via USB. Only 31 out of 32 are measured by the cushion sensor, and the last one is dummy data. 31 data are received in the order of the cell number of the cushion sensor. (See page 5 for the cell number of the cushion sensor) The 31 read data is stored in cellValues[] array.

2) Reordering the buffer cellValues[]

The cushion sensor consists of 3 rows, and each row has a buffer. The 31 values in the receive buffer cellValues[] are the data of the entire cushion sensor, which is divided into 3 row buffers and stored.



Receiving buffer

cellValues[31]

Row Buffer (member of SeatDataView class)

seatValues_Row0 [6]

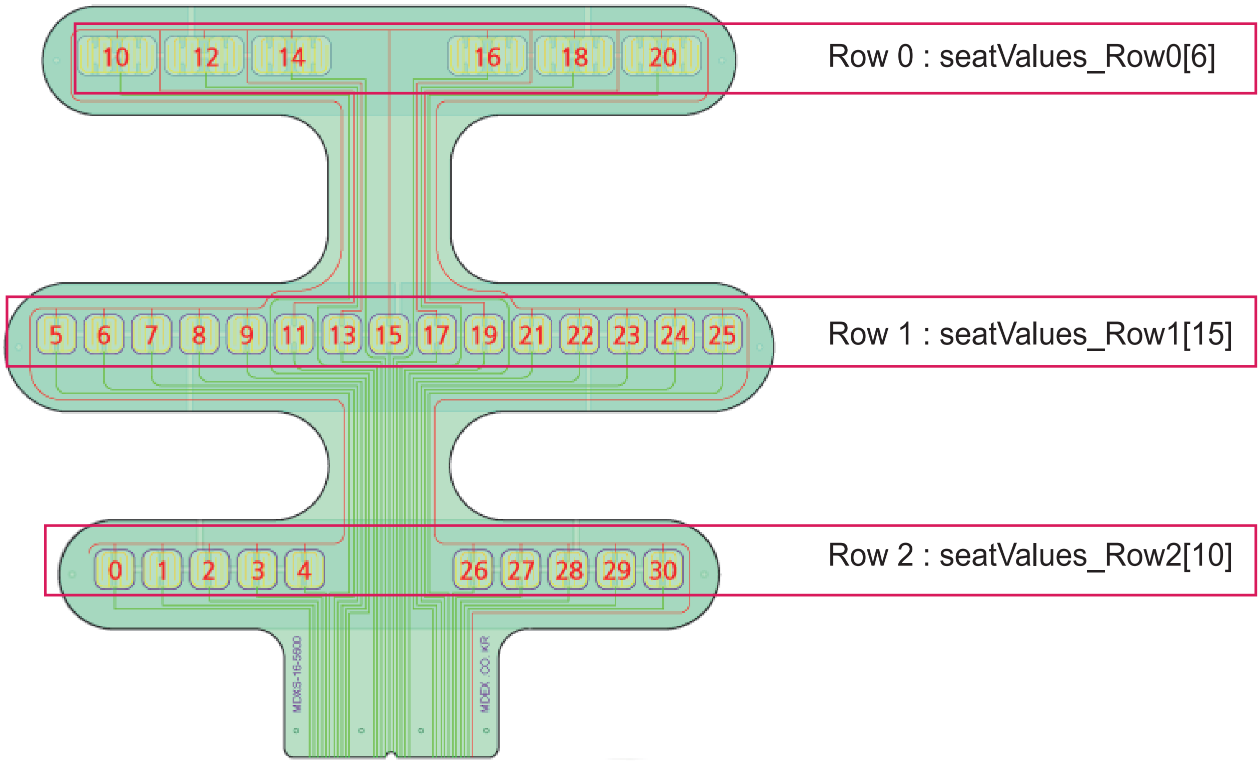
seatValues_Row1 [15]

seatValues_Row2 [10]

Reordering to 3 rows

Data transfer order

Buffer of 3 Rows

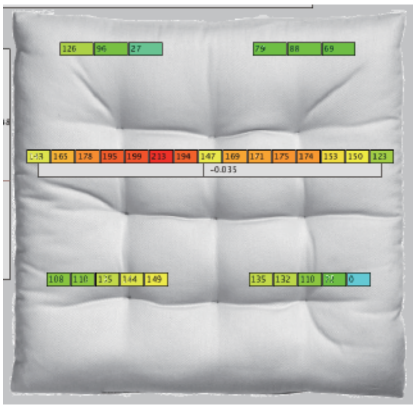

10
12
14
16
18
20
Row 0 : seatValues_Row0[6]
5
6
7
8
9
11
13
15
17
19
21
22
23
24
25
Row 1 : seatValues_Row1[15]
0
1
2
3
4
26
27
28
29
30
Row 2 : seatValues_Row2[10]
MDXS-16-5500
MDEX CO. KR

# Visualization SW - 2. Color Cells

Model : Cushion sensor coding kit

## UI-Set the coordinates of 31 color cells (rect_pos.csv)

### Color Cell UI-Change background image.

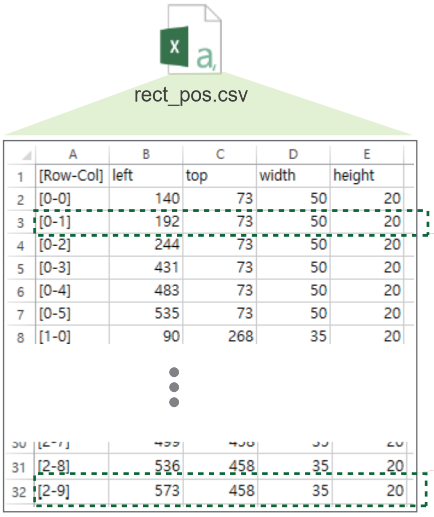


You can change the background image of the app. Please put the image file in the image folder.

Just change the file name parameter in the setup() function.
imgSensor = loadImage(../image/new image file name.png);

### Color Cell UI-Positioning

The coordinates of the 31 color cell UI rectangles are read from the rect_pos.csv file in the image folder. Changing the number in the CSV file changes the cell's position. If you do not intend to change the coordinates, you can skip this paragraph.


rect_pos.csv

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | [Row-Col] | left | top | width | height |
| 2 | [0-0] | 140 | 73 | 50 | 20 |
| 3 | [0-1] | 192 | 73 | 50 | 20 |
| 4 | [0-2] | 244 | 73 | 50 | 20 |
| 5 | [0-3] | 431 | 73 | 50 | 20 |
| 6 | [0-4] | 483 | 73 | 50 | 20 |
| 7 | [0-5] | 535 | 73 | 50 | 20 |
| 8 | [1-0] | 90 | 268 | 35 | 20 |
| ⋮ | | | | | |
| 30 | [2-7] | 499 | 458 | 35 | 20 |
| 31 | [2-8] | 536 | 458 | 35 | 20 |
| 32 | [2-9] | 573 | 458 | 35 | 20 |

[0-1] : Row 0, Column 1

Coordinate (192, 73)

Height = 20

Width = 50

[2-9] : Row 2, Column 9

Coordinate (573, 458)

Height = 20

Width = 35

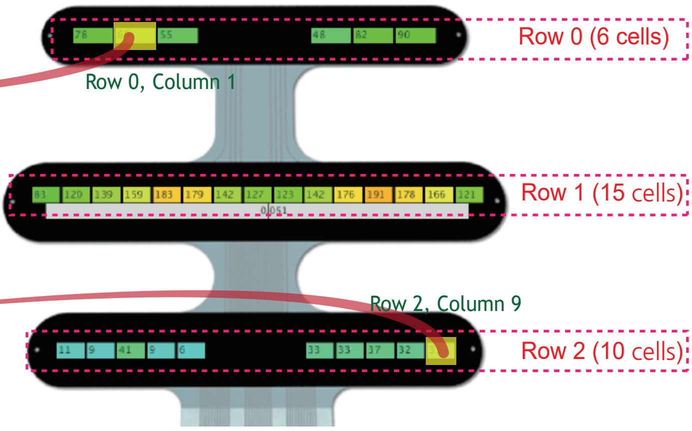


Figure) The 31 color cell squares are grouped into 3 rows.

### Color Cell UI - Color Settings

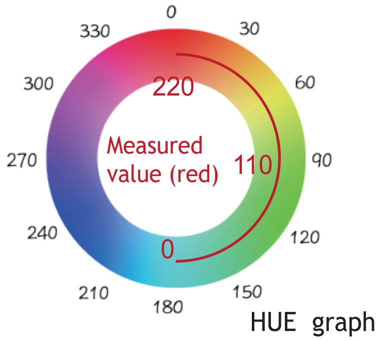


Figure) HUE graph of HSB colormap. In the example, the range of the pressure value (0~220) and the range of the HUE value (180~0) were matched.

Function : colorMap in SeatDataView class

If the pressure value is 0, then HUE is 180.
If the pressure value is 220, then HUE is 0.

Convert the measured value to HUE value with the formula below.

HUE = 180 - (Measured value * 180 / 220)

For example, if the measured value is 40?
HUE = 180 - (40 x 180 / 220 ) = 148

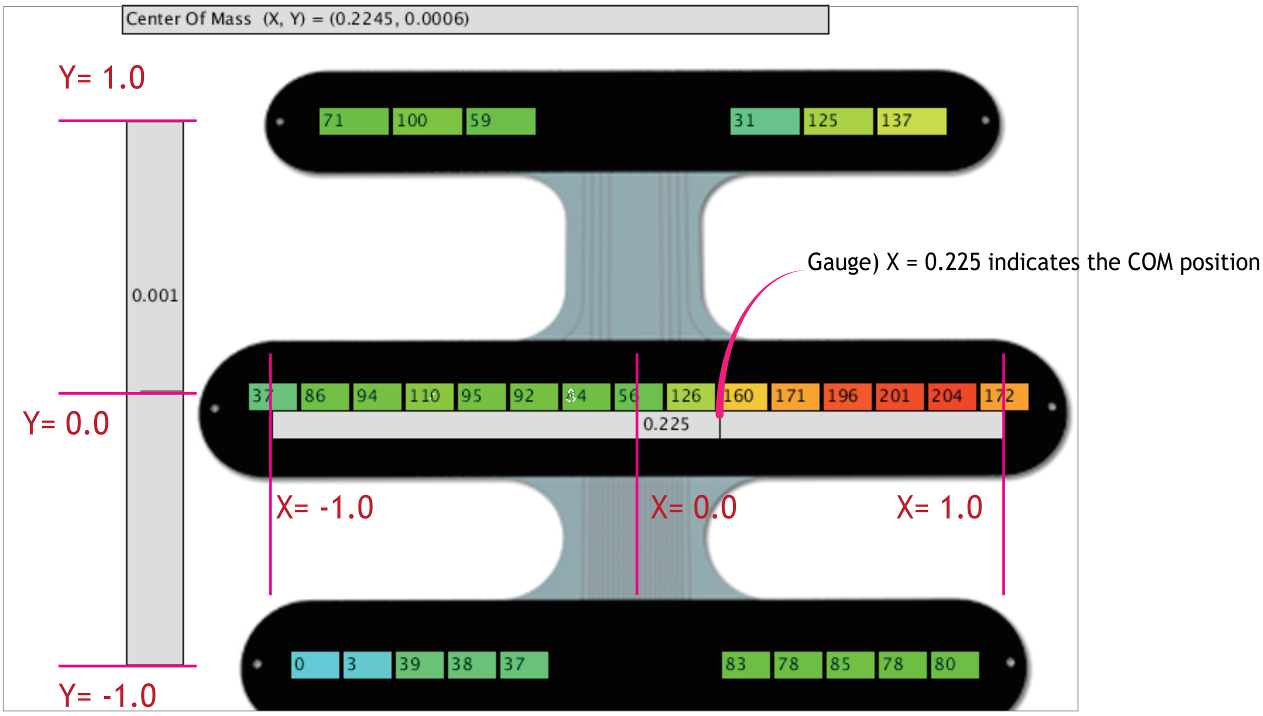
Exception) If the measured value is 220 or higher, set HUE to 0.

# Visualization SW - 3. Center of Mass

Model : Cushion sensor coding kit

## Calculation and display of the center of mass value.

The relative coordinates of the center of mass are displayed in the top bar of the screen. In addition, gauges are displayed on the left and center of the screen.
The coordinate system of relative coordinates is indicated in red below. (X=-1.0, X=0.0, X=+1.0, and Y=-1.0, Y=0.0, Y=+1.0)
Take the X axis for example. X when tilted to the far left is -1.0, and 1.0 when tilted to the far right.



Center Of Mass (X, Y) = (0.2245, 0.0006)

Gauge) X = 0.225 indicates the COM position

### Calculation of COM in X direction

Formula for COM

$$center\ of\ mass = \frac{sum\ of\ all\ (position \times mass)}{sum\ of\ all\ masses}$$

X-direction Calculating

$$X = \frac{\sum_i^n m_i x_i}{\sum_i^n m_i} = \frac{m_1 x_1 + m_2 x_2 + \cdots + m_n x_n}{m_1 + m_2 + \cdots + m_n}$$

$x_i$ : x-coordinate value
$m_i$ : mass, pressure value

Function name in X-direction COM calculation: : calcCOM_X() (in SeatDataView class)

```
for (int i = 0; i < cellNum_Row1; i++) {
  float cell_pos_x = -1.0f + i * pitch_x;
  sum_weight_pos += seatValues_Row1[i] * cell_pos_x;

  sum_weight += seatValues_Row1[i];
}

com_x = sum_weight_pos / sum_weight;
```

seatValues_Row1[i] : pressure value
cell_pos_x : x coordinate value

It is possible to calculate the center of mass in the Y direction in a similar way.

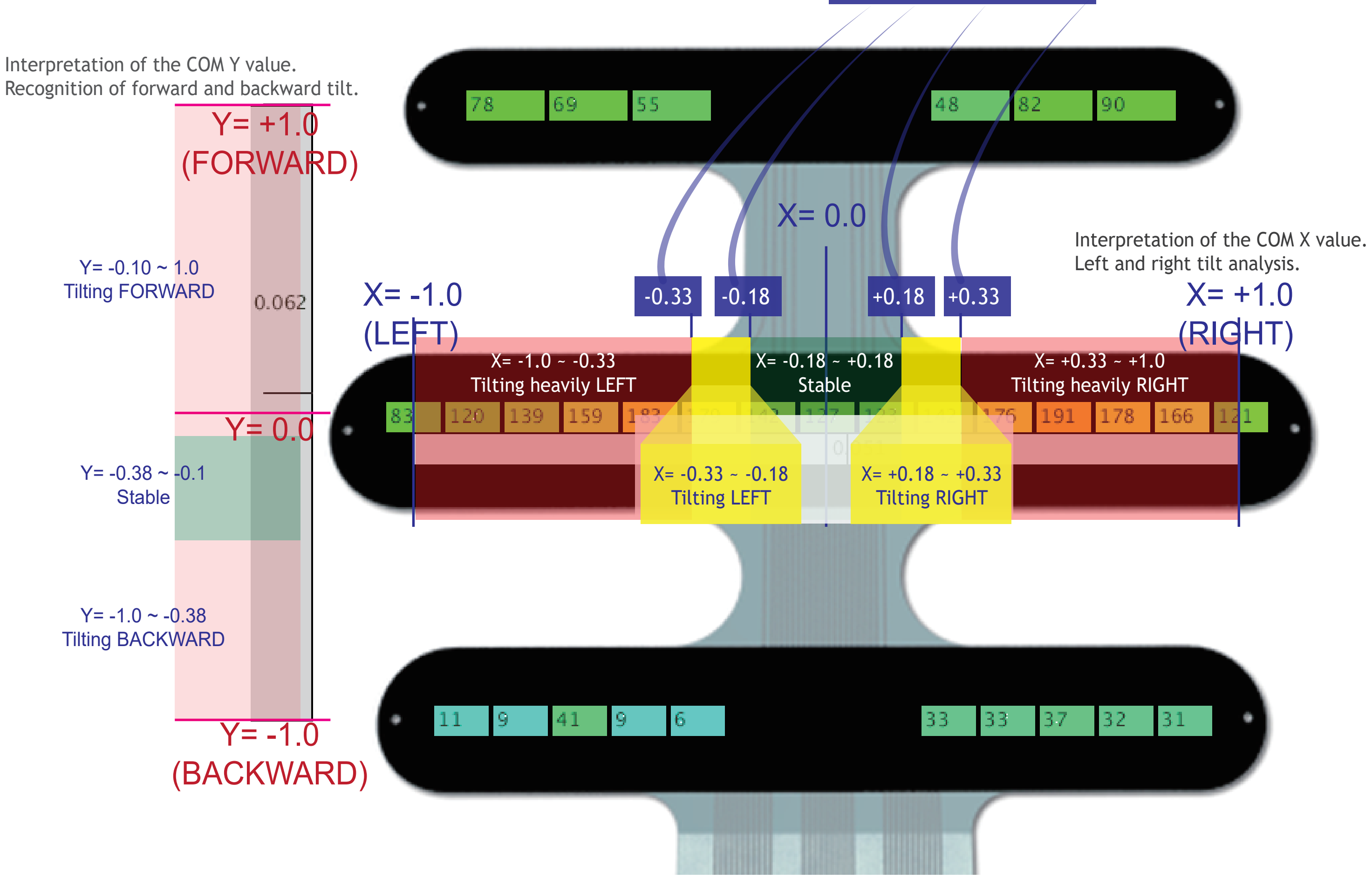
# Visualization SW - 4. Aware 'Tilt' on COM

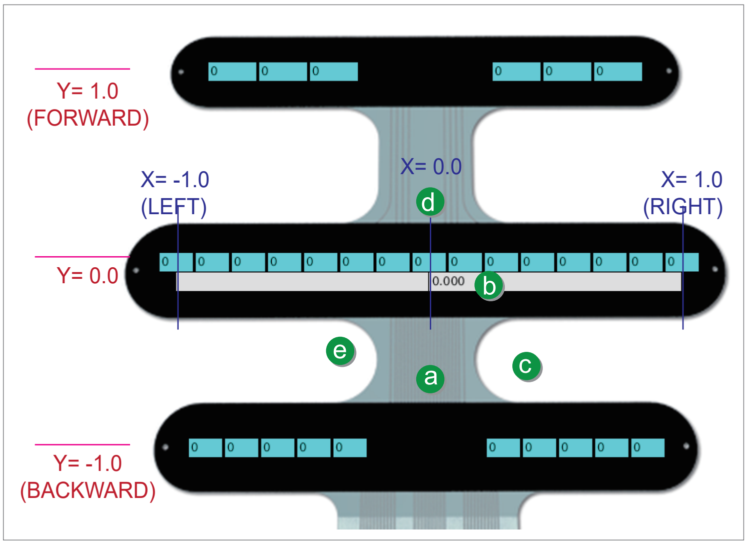
Model : Cushion sensor coding kit

## Use of the center of mass calculation results (see previous page)

### Setting the center of mass value and threshold

If the center of mass value is not (0, 0), that is, a value greater than or less than 0, it means that the occupant's load is tilted in either direction.
The center of mass value is -1.0~+1.0, and by specifying the threshold value between these, the tendency of sitting posture can be defined.
In the example in the figure below, the thresholds for tilting in the X direction are set to -0.33, -0.18, +0.18, and +0.33.





### Additional Tilting Recognition Examples

- a If COM is (0, -0.59) : Tilt Backward
- b If COM is (+0.26, -0.17): Tilt Right
- c If COM is (+0.38, -0.54): 'Tilt right heavily', and 'Tilt backward'.
- d If COM is (0, +0.35): Tilt forward.
- e If COM is (-0.32, -0.43) : 'Tilt left' and 'Tilt Backward'

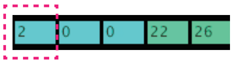
There may be errors in the recognition/discrimination posture with this cushion sensor.
Factors affecting: the shape of the seated person, the shape of the chair/chshion, the material, etc.

# Visualization SW - 5. Invalid / Fixed cell

Model : Cushion sensor coding kit

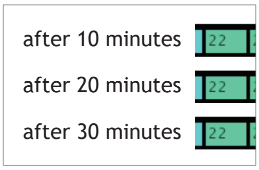
## Validity filtering

### Invalid cell



Situation-Values can be measured without being pressed.
1) When electrical noise is mixed during measurement
2) When the sensor is slightly bent because the floor is not flat. In this case, the normal output value is 5 or less low.
Recommended-Values below 5 are considered invalid cells

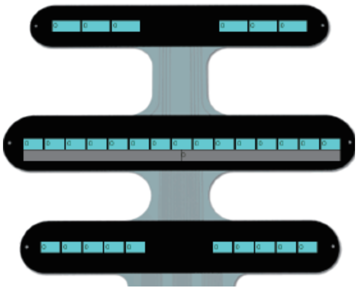
### Fixed cell


after 10 minutes 22
after 20 minutes 22
after 30 minutes 22

Cell with very small changes in measured values over several minutes are called fixed cell.
You may suspect that you put something on it, not a person.

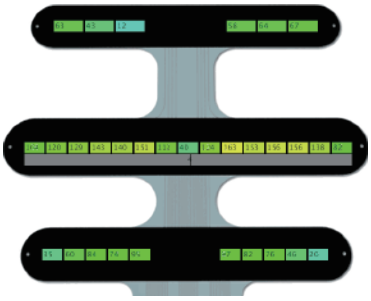
## Use the sum of 31 cell values

### Recognizing - Empty



Recommended - If all the sum of the measurement values of 31 cells is 50 or less, it may be judged as empty.

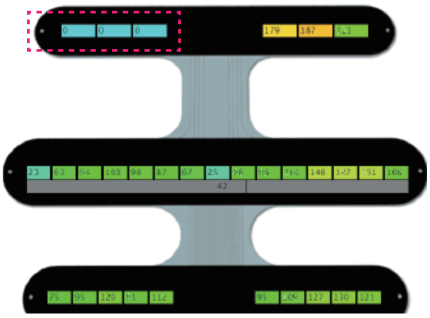
### Recognizing - Sitting state



Recommended-If the sum of all cell measurements is 200 or more, it can be judged as sitting
Caution-If all 31 cells are'fixed cells' for 20 minutes, non-human objects (eg, rice bags) may be placed on the sensor.

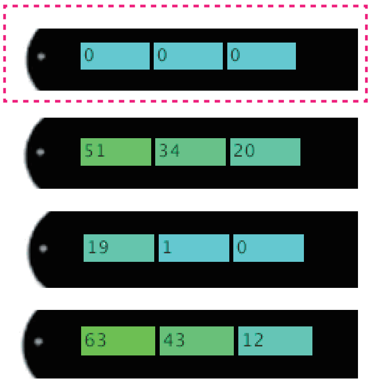
## Using invalid cells for posture recognition

### Recognizing - Legs crossed



Recommendation-6 cells in row 0 are divided into 3 cells on the left and 3 cells on the right.
If there are two or more invalid cells among the three cells on either side, the legs can be recognized to be crossed.

### Recognizing - Shaking leg



Release pressure
Apply pressure
Release pressure
Apply pressure

If the pressure is repeated for seconds for the three left cells in row 0, as shown, the left leg is likely to be shaking.
Recommendation-If the sum of the measured values of the three cells on one side of the left/right side in row 0 frequently fluctuates, it is decided to shake the leg.

There may be errors in the recognition/discrimination posture with this cushion sensor. Factors affecting: the shape of the seated person, the shape of the chair/chshion, the material, etc.

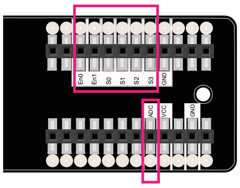
# Using the idle pins of Arduino

Model : Cushion sensor coding kit

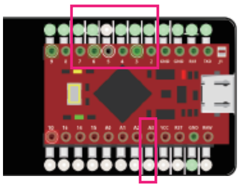
## Find out which idle pins are available on the shield

Arduino boards usually have multiple external pins. Except for basic pins such as VCC and GND, we have around 20 function pins.
Except for the 7 pins (6 GPIO and 1 ADC) used to measure the cushion sensor, the remaining pins may be used for other purposes.

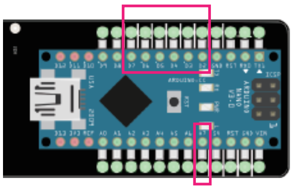
### 7 pins for measuring the cushion sensor-marked with a red square. Not usable for other purposes



Shield

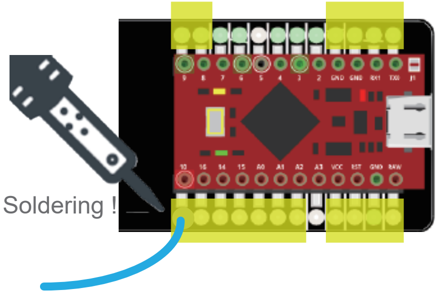


Arduino ProMicro



Arduino NANO

### Idle pins that can be used for other purposes. -Yellow square

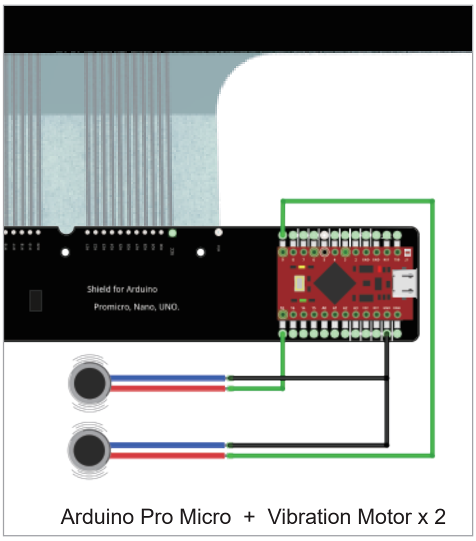


Solder the wire to the idle pin to connect other parts.

## Use idle pins-connect other components, SPI

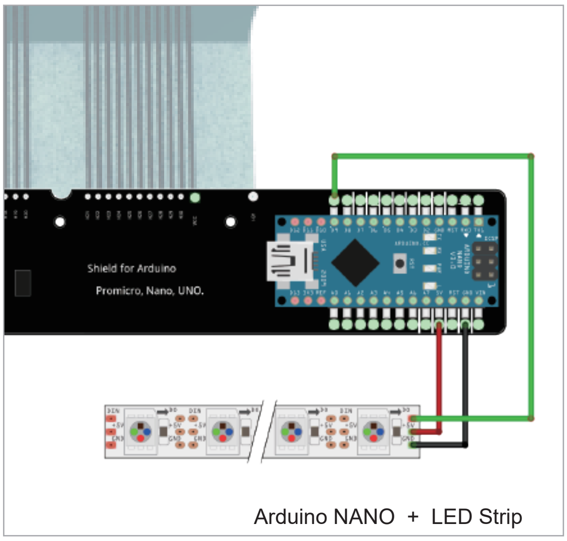
You can connect various parts on the market.

Example) Vibration motor (PWM, VCC, GND)



Arduino Pro Micro + Vibration Motor x 2

Example) ws2812 NeoPixel LED strip connection (PWM, VCC, GND)



Arduino NANO + LED Strip

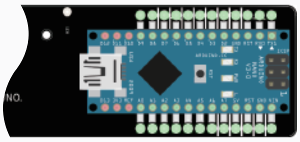
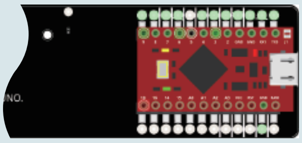
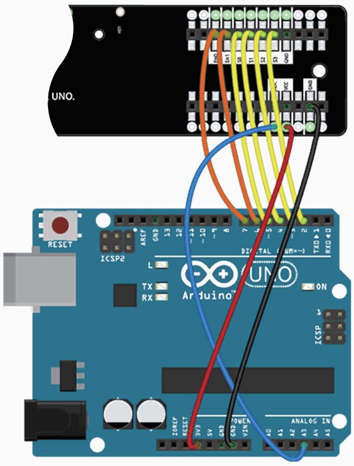
The method of using the parts used in the example above can be changed, so it is recommended to apply the parts after checking the manufacturer's documentation.

# More measurement circuits

Model : Cushion sensor coding kit

## Arduino series

Various Arduino boards can measure by connecting them to the cushion sensor shield. Low-cost Arduino compatible boards are also OK for measurement. Connect 6 GPIO pins and 1 ADC pin, and specify these 7 pin numbers in the definition part of the Arduino source code.
(If you are not sure about the pin setting in the source code, please contact us. s@mdex.co.kr)

| Characteristic | Arduino Model Name | How to connect |
|---|---|---|
| BLE Wi Fi | NANO 33 BLE Sense with headers (Connectivity: Bluetooth BLE)<br>NANO 33 IOT (Connectivity : Bluetooth BLE , WIFI )<br>NANO (Connectivity : USB)<br>In the case of BLE or WIFI board, power can be supplied from the supplementary battery through the USB terminal. |  |
| HID | ProMicro (HID is available)<br>What is HID? - Human Interface Device.<br>Devices that can generate keyboard/mouse events |  |
| UNO | UNO- Wire connection with shield.<br>Connect 6 GPIOs and 1 ADC.<br>VCC and GND must also be connected.<br>Other models are available by connecting wires<br>--> Most models such as Arduino UNO, MEGA, Micro, Due, etc. |  |

## Measurement other than Arduino

Examples) Micro:bit, ESP32 , etc.

Connect 6 pins of GPIO and 1 pin of ADC with cushion sensor shield.
(Including VCC and GND)

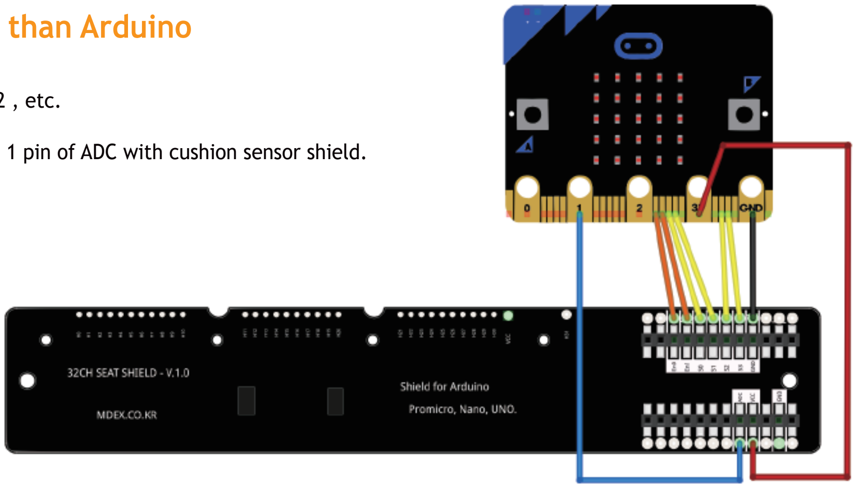


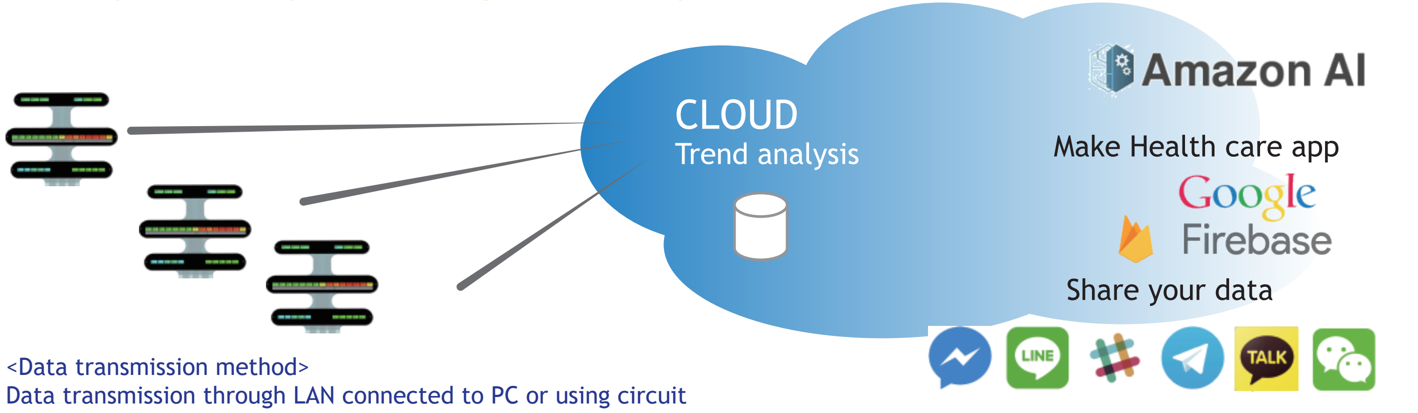
Figure)
Connection of cushion sensor shield and Micro:bit board

# Challenges- Utilizing Measurement Data

## Model : Cushion sensor coding kit

So far, we have seen how to measure data with the cushion sensor kit. We discussed some examples of possible tasks as shown below with experts. Use your imagination and challenge various tasks. (Project consultation: s@mdex.co.kr)

### Cloud-big data storage statistics, posture recognition AI



CLOUD
Trend analysis

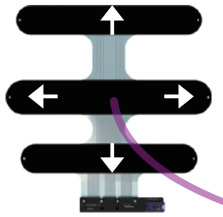
Make Health care app

Share your data

<Data transmission method>
Data transmission through LAN connected to PC or using circuit with WIFI function (e.g. Arduino NANO IOT)

### Link Sensor data to Experience-based content

According to the center of mass value of the cushion sensor, the application outputs the joystick direction key event.
That is, if the body is tilted to the left, a left key event can be generated.
Tilt back then generates the down key event.

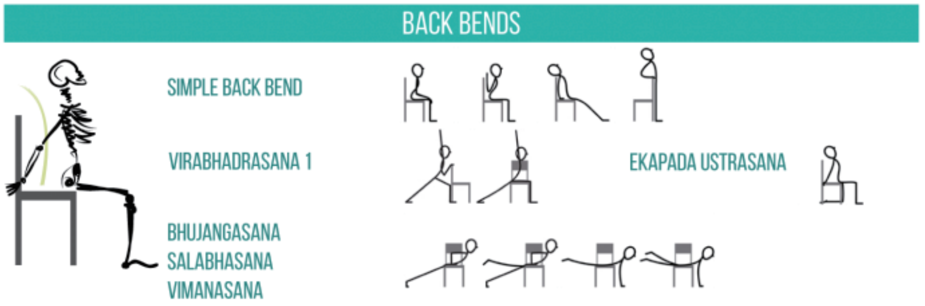


<Key event generation>
With Arduino ProMicro, you can generate key events.
Provide example code in the blog.

Key events utilizing in VR or FPS games.
(E.g. arrow keys, lying down, jumping, changing equipment, etc.)

### Chair yoga

There are several ways to stretch while sitting in a chair. (Search 'chair yoga' in Google)
You can check whether the yoga pose is properly implemented by measuring the cushion sensor. If you add a scheduler here, you can create a cushion yoga coaching application.



# Cushion sensor reliability

## Model : Cushion sensor coding kit

The performance of the specifications below is not guaranteed for all manufactured products. The information in this document is a reference for a rough overview of product performance.

## Typical Performance

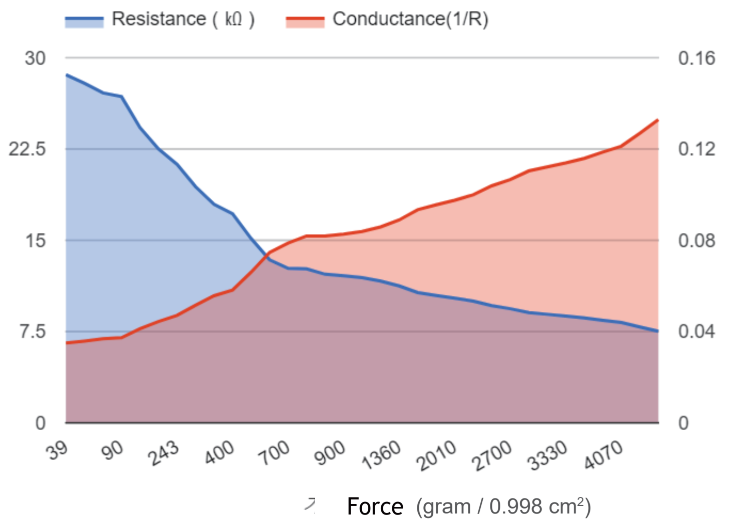
| Unit | Description |
|---|---|
| Response time | < 10 μsec |
| Operating temp. | -20℃ ~ 50℃ |
| Storage temp. | -30℃ ~ 60℃ |
| Storage humidity | ≤ 90% |
| Durability | 2,000,000 stroke (100g) or over 500,000 stroke (150g) |
| Drift | < 5% per logarithmic time scale by constant load of 100g |
| Electric crosstalk (noise) | None |
| Power consumption | Consumes only while operating. Typically around 5mA, and maximum 20mA. |
| Resistance output range | ∞ ~ 200Ω , Infinity at no force. |
| Sensing range (Single cell) | 20 gf ~ 4 kgf , while 30 kgf ~ 150 kgf by whole 31 cells sensor. |

All the characteristics of temerature and humidity in this page are the result of 96 hours test.

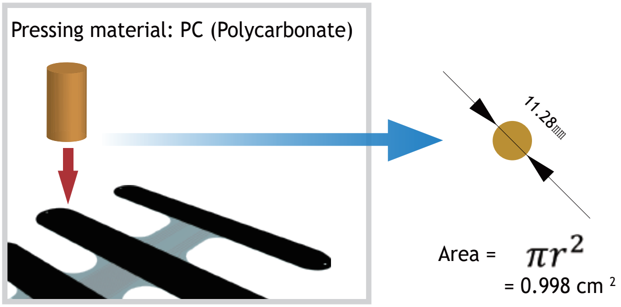
## Tolerance

| Unit | Description |
|---|---|
| Mechanical tolerance | ≤ 150μm |
| Temperature influence | ≤ 15% |
| Humidity influence | ≤ 50% [1)] |
| Output tolerance | Max 20% |

## Graph) Force to Resistance (and Conductance = 1/R)



### How to get this graph..



2) If the material (rigidity/ductility) and the cross-sectional area of the area where the sensor is pressed are different, the slope of the graph will change. Therefore, it is not recommended to use the F-R graph on the left for load conversion purposes (balance). Please consider it as a reference only.

1) The sensor output value may change due to temperature and humidity. If the temperature and humidity affect the entire 31 cells of the cushion sensor, the output degradation of the 31 cells occurs at an almost uniform level. In this case, the degree of damage to the relative comparison function of the load is not large. Therefore, the impact on the “seat center of mass perception” function using relative comparison is much more limited than the deviation in this table.

# Precautions

## Model : Cushion sensor coding kit

## Notes on use

Please note that the following items may cause sensor damage/malfunction.

Fold the sensor tightly.
Use the sensor in a severely squeezed state. (The neck may bend)
Press the sensor with a pointed object. (E.g. awls, nails, high heels, etc.)
Press the sensor in the lateral direction. (Example: Press while rubbing the sensor.)
Special care is required when soldering terminal parts. Long soldering times can cause damage. It should be worn to a low temperature as soon as possible.
Burn the sensor. Or sparks.
Immerse the sensor in water.
Store the sensor in a sealed state close to the adhesive liquid (bond, grease, etc.).

## Precautions when handling/storing the sensor-chemicals

Although the composition of this sensor is known to be chemically stable, there are some caveats. The following are known to be vulnerable to sensors:

| Material | chemical resistance |
|---|---|
| Hydrocarbons, mainly occur when fossil fuels such as petroleum are burned | Vulnerable |
| Ketones, Acetone | Slightly vulnerable(long-term deformation) |
| weak alkali | Slightly vulnerable(long-term deformation) |
| OIL, Gasoline, Grease | Slightly vulnerable(long-term deformation) |
| kerosene | No data |
| Methanol | Resistant |
| Ethanol | Resistant |
| Isopropanol | Resistant |
| Weak acid | Resistant |

<How to use the bond>
* It is necessary to check whether the bond, acetone, and hardener have dried after use.
In addition to the substances mentioned above, there may be other chemicals that can desensitize.
Even though the material is mentioned as 'Resistant' above, it is recommended to test various chemicals exposed to the sensor during the customer's assembly process or delivery/storage to ensure that it is safe.

Sealed-If the sensor is sealed with other chemicals (closed-not just a complete seal, but also a plastic packaging), a complex reaction may occur, so it is recommended to conduct a reaction test.
Also, the material listed above does not always break the sensor. If the sensor is exposed to the chemical for a few seconds or in an unsealed environment, it is less affected by the chemical.

## Aging tendency

Repeated use of this product will cause usability aging.

### Aging-if the sensor becomes more sensitive

If the film, the material of the sensor's top plate, is repeatedly subjected to force, it loses its elasticity. As a result, it acts as if a higher load is applied than it actually is. For example, if 1 million is applied with 150 gf, the sensor may be sensitive to around 15%.
After repeated pressurization or very strong pressurization, the sensor top plate may come into contact with the bottom plate. In this case, the output is generated even in the standby state without pressing the sensor.
Even if the output occurs in the standby state, the ADC output value is very low compared to the output value by actual pressure, so it can be distinguished through SW calibration. Usually the output in the aged standby state is 5 or less.

### Aging-if you become more insensitive

If the surface inside the sensor is damaged by exposure to chemicals, high temperature/high humidity for a long time, the sensitivity may become dull.

## Always pressurized - Drift

After applying a load to the sensor, if the time passes without removing the load, the measured resistance value gradually decreases.
Link to the drift video-https://youtu.be/WDrCZ1VcZBg (from 637 ohms to 624 ohms in 1 hour)

After a few hours or more, the resistance almost falls to the bottom of the output. This phenomenon is independent of product aging, and will recover as soon as pressure is removed. The sensor should be designed so that no load is applied to the sensor while idle.
Even in the cushion sensor, drift occurs in a small amount. However, the drift occurs very slowly because the occupants have small movements.